

TUGAS AKHIR
TIMBANGAN DIGITAL BERBASIS ARDUINO UNO DENGAN
OUTPUT SUARA

*Sebagai Salah Satu Syarat Untuk Menyelesaikan
Program Studi Diploma III Jurusan Teknik Elektronika*



Oleh :

MUHAMMAD RIDWAN
3103151021

JURUSAN TEKNIK ELEKTRO
POLITEKNIK NEGERI BENGKALIS
TAHUN 2018

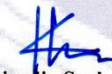
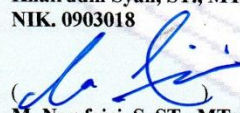
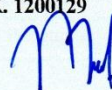

HALAMAN PENGESAHAN
TIMBANGAN DIGITAL BERBASIS ARDUINO UNO
DENGAN OUTPUT SUARA

*Sebagai salah satu syarat untuk menyelesaikan
Program Studi Diploma III Jurusan Teknik Elektro*


Oleh:

MUHAMMAD RIDWAN
3103151021

*Disetujui oleh Tim Penguji Tugas Akhir : Tanggal Ujian : 14 Agustus 2018
Periode Wisuda : XV*


- ()
1. **Khairudin Syah, ST., MT** (Pembimbing)
NIK. 0903018
- ()
2. **M. Nur faizi, S. ST., MT** (Penguji 1)
NIK. 1200129
- ()
3. **Marzuarman, S. Si., MT** (Penguji 2)
NIK. 1200153
- ()
4. **Syaiful Amri, S. ST., MT** (Penguji 3)
NIK. 0909134

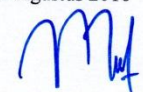
Bengkalis, 24 Oktober 2018
Ketua Program Studi DIII Teknik Elektronika

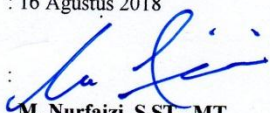

AGUSTIAWAN, S.ST., MT
NIP.198508012015041005

LEMBAR PENGESAHAN

Kami dengan sebenarnya menyatakan bahwa, kami telah membaca keseluruhan dari Tugas Akhir ini dan kami berpendapat bahwa Tugas Akhir ini layak dan memenuhi syarat untuk memperoleh gelar Ahli Madya.

Tanda tangan : 
Nama Penguji I : **Syaiful Amri, S.ST., MT**
Tanggal Pengujian : 16 Agustus 2018

Tanda tangan : 
Nama Penguji II : **Marzuarman, S.Si., MT**
Tanggal Pengujian : 16 Agustus 2018

Tanda tangan : 
Nama Penguji III : **M. Nurfaizi, S.ST., MT**
Tanggal Pengujian : 16 Agustus 2018

HALAMAN PERNYATAAN

Saya menyatakan dengan sesungguhnya bahwa Tugas Akhir ini adalah hasil karya saya dan tidak terdapat karya yang pernah dilakukan untuk memperoleh gelar ahli madya di perguruan tinggi, dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau dipublikasikan oleh orang lain, kecuali yang secara tertulis disebutkan sumbernya dalam naskah dan dalam daftar pustaka

Bengkalis, 3 Oktober 2018

Penulis



Muhammad Ridwan

HALAMAN PERSEMBAHAN

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Sujud syukurku kusembahkan kepada Tuhan yang Maha Agung nan Maha Tinggi nan Maha Adil nan Maha Penyayang, atas takdirMu telah Kau jadikan aku manusia yang senantiasa berfikir, berilmu, beriman dan bersabar dalam menjalani kehidupan ini. Semoga keberhasilan ini menjadi satu langkah awal bagiku untuk meraih cita-cita besarku.

Kepada kedua orang tua adik dan semua keluargakuterima kasih atas semua dukungan dan doa selama ini. Terima kasih juga untuk teman-teman seperjuangan saya yaitu T.E VI A dan T.E VI B, tanpa kalian semua saya mungkin tidak akan menyelesaikan Tugas Akhir ini.

Kepada bapak Khairudin Syah, S.ST., MT selaku dosen pembimbing saya yang telah memberikan pengarahan, bantuan, serta ilmu untuk saya, sehingga saya dapat menyelesaikan Tugas Akhir saya. Terima kasih kepada seluruh dosen dan staf yang mengajar atas semua ilmu yang telah berikan.

“TANPA DOA KALIAN SAYA TIDAK AKAN MENJADI ORANG SUKSES”

KATA PENGANTAR

Bismillahirrohmanirrohiim.

Assalamu'alaikum. Wr. Wb.

Alhamdulillahirobbil'alamin, segala puji dan syukur kepada Allah SWT atas rahmat dan karunianya, sholawat dan salam buat junjungan alam kepada Nabi Muhammad SAW yang telah membawa umat dari alam kejahilian menuju alam yang penuh dengan ilmu pengetahuan sehingga penulis dapat menyelesaikan penulisan Tugas Akhir (TA) yang berjudul “ Timbangan Digital Berbasis Arduino Uno Dengan Output Suara”.

Maksud dari penulisan Tugas Akhir (TA) ini adalah sebagai salah satu syarat untuk memperoleh gelar Diploma III (D3) Jurusan Teknik Elektro Politeknik Negeri Bengkalis. Penulis menyadari masih banyak kekurangan dalam penyusunan laporan tugas akhir ini dari kesempurnaan, oleh karena itu penulis senantiasa mengharapkan kritik dan saran yang bersifat membangun guna untuk perbaikan dan kelengkapan tugas akhir ini. Pembuatan Tugas Akhir ini, penulis banyak sekali mendapatkan bantuan dari beberapa pihak. Untuk itu di sini penulis ingin mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Kedua orang tua saya yaitu ibunda Yusna Hermawati dan ayahanda Suriyanto yang telah banyak memberi semangat dan doa kepada saya.
2. Adik dan keluarga saya yang telah memberi semangat dalam penyusunan Tugas Akhir selama ini.
3. Bapak Ir. Muhammad Milchan, MT, selaku Direktur Politeknik Negeri Bengkalis.
4. Bapak Wan M. Faisal, ST.,MT, selaku Ketua Jurusan Teknik Elektro.
5. Bapak Agustiawan,S. ST., MT, selaku Ketua Program Studi D3 Teknik Elektronika.
6. Bapak Khairudin Syah, ST., MT, selaku pembimbing.
7. Bapak Marzuarman, S.Si., MT, selaku Koordinator Tugas Akhir ini.

8. Bapak dan Ibu dosen Politeknik Negeri Bengkalis yang telah memberikan sebagian ilmu kepada penulis.
9. Teman-teman seperjuangan yang ikut serta membantu penulis dalam menyelesaikan penulisan laporan Tugas Akhir ini.

Akhirnya, penulis mengharapkan semoga penulisan laporan tugas akhir ini dapat bermanfaat khususnya bagi penulis dan mahasiswa jurusan Teknik Elektro Politeknik Negeri Bengkalis, maupun pembaca umumnya.

Bengkalis, 3 Oktober 2018

Penulis

TIMBANGAN DIGITAL BERBASIS ARDUINO DENGAN OUTPUT SUARA

Nama Mahasiswa : Muhammad Ridwan
NIM : 3103151021
Pembimbing : Khairudin Syah, ST., MT

ABSTRAK

Perkembangan teknologi pada alat ukur, menyebabkan jenis alat ukur massa atau alat timbangan semakin bertambah sesuai dengan fungsinya masing-masing. Salah satu bentuk perkembangan teknologi terhadap alat ukur massa terdapat pada alat timbangan digital yang berkembang menjadi alat timbangan suara. Tujuan dari penelitian ini adalah membuat timbangan berbasis arduino dengan *output* LCD dan suara yang mempunyai fungsi untuk membantu tuna netra yang ingin berdagang tanpa melihat LCD karena ada *output* suara yang dapat di dengarnya. Karena pengukuran berat sering menjadi hal penting bagi para penjual untuk memastikan harga yang sesuai dengan hasil timbangan yang diukur. Beberapa kendala yang dialami para penjual adalah menyetel ulang posisi jarum penunjuk supaya menjadi 0, baru kemudian mereka melakukan timbang menimbang. Berdasarkan hasil uji coba alat didapatkan nilai persentase eror alat adalah 0,208%. Dari hasil persentase eror dapat dinyatakan baik dengan nilai rata-rata 20,8%. Berdasarkan hasil uji coba, dapat disimpulkan bahwa pembuatan alat timbangan yang telah diuji menyatakan bahwa sistem kerja alat dinyatakan baik dan layak, akan tetapi alat ini belum dapat dipasarkan karena hasil yang didapat belum mencapai/akurat sepenuhnya. Sehingga perlu di kembangkan lagi.

Kata kunci : Arduino Uno, Load Cell, HX711, MP3 Serial Player, Speker

DIGITAL SCALE BASED ON ARDUINO WITH SOUND OUTPUT

*Student Name : Muhammad Ridwan
NIM : 3103151021
Advisor : KhairudinSyah, ST., MT*

ABSTRACT

Technology developments in measuring instruments, measurement of types of mass measuring devices or scales that increase according to their respective functions. One form of technological progress towards measuring devices on digital scales that developed into sound scales. The purpose of this study is to make an arduino-based scale with LCD output and sound that has a function to help blind people who want to trade without looking at the LCD because there is a sound output that can be heard. Thus, weight measurement is very important for sellers to ensure prices are in line with the results of the scales that are measured. Some of the leaders who carry out the task are to reset the position of the pointer to 0, and then they weigh weighing.

Based on the results of the trial tool to determine the percentage of tool error is 0.208%. The percentage results can be given with an average average of 20.8%. Based on the results of the study, it can be concluded that the manufacturing of weighing devices that have been used states that the work system of the tool is good and feasible, but this tool has not yet been marketed because the results obtained have not yet reached full accuracy. Need to develop again.

Keywords: Arduino Uno, Load Cell, HX711, MP3 Serial Player, Speker.

DAFTAR ISI

	Halaman
HALAMAN JUDUL	
HALAMAN PENGESAHAN	i
LEMBAR PENGESAHAN	ii
HALAMAN PERNYATAAN	iii
HALAMAN PERSEMBAHAN	iv
KATA PENGANTAR	v
ABSTRAK	vii
ABSTRACT	viii
DAFTAR ISI	ix
DAFTAR GAMBAR	xi
DAFTAR TABEL	xiii
BAB I PENDAHULUAN	
1.1 Latar Belakang	1
1.2 Rumusan Masalah	1
1.3 Batasan Masalah	2
1.4 Tujuan Penelitian	2
1.5 Manfaat Penelitian	2
1.6 Sistematika Penulisan	3
BAB II TINJAUAN PUSTAKA	
2.1 Penelitian Terdahulu	4
2.2 Landasan Teori	4
2.3 Komponen yang digunakan	5
2.3.1 Arduino uno	5
2.3.2 Load cell	6
2.3.3 IC HX711	7
2.3.4 LCD (<i>liquid cristal display</i>)	9

2.3.5 <i>Power supply</i>	11
2.3.6 <i>Df player</i>	11
2.3.7 <i>Transformator</i>	12
2.3.8 <i>Speaker 5 watt</i>	14
2.3.9 <i>Handphone</i>	15
2.3.10 <i>Kabel jumper</i>	16
2.3.11 <i>Resistor</i>	16
2.3.12 <i>Kapasitor</i>	18
2.3.13 <i>Dioda</i>	20
BAB III METODOLOGI PENELITIAN	
3.1 <i>Tinjauan Umum</i>	22
3.2 <i>Metode Penelitian</i>	23
3.3 <i>Studi literatur</i>	24
3.4 <i>Blok diagram Sistem</i>	24
3.5 <i>Algoritma pemrograman</i>	26
3.6 <i>Perancangan hardware</i>	27
3.7 <i>Perancangan software</i>	28
3.7.1 <i>Pemrograman arduino</i>	28
3.8 <i>Rangkaian prototype</i>	29
BAB IV HASIL PENELITIAN DAN ANALISA	
4.1 <i>Hasil perancangan</i>	30
4.2 <i>Pengujian load cell</i>	30
4.3 <i>Pengujian power supply</i>	31
4.4 <i>Pengujian LCD</i>	32
4.5 <i>Pengujian alat keseluruhan</i>	32
BAB 5 PENUTUP	
5.1 <i>Kesimpulan</i>	36
5.2 <i>Saran</i>	37
DAFTAR PUSTAKA	38
LAMPIRAN	
BIOGRAFI PENULIS	

DAFTAR GAMBAR

Gambar 2.1 Arduino Uno.....	5
Gambar 2.2 Atmega 328 Pin Mapping	5
Gambar 2.3 Load Cell.....	6
Gambar 2.4 IC HX711	8
Gambar 2.5 LCD.....	9
Gambar 2.6 Df Player	11
Gambar 2.7 Transformator.....	12
Gambar 2.8 Speker.....	15
Gambar 2.9 <i>Handphone</i>	16
Gambar 2.10 Kabel Jumper	16
Gambar 2.11 Resistor.....	17
Gambar 2.12 Kapasitor	19
Gambar 2.13 Dioda	20
Gambar 3.1 Metode Penelitian.....	23
Gambar 3.2 Blok diagram	25
Gambar 3.2 Algoritma pemrograman	26
Gambar 3.3 Rangkaian <i>Hardware</i>	26
Gambar 3.4 Pemrograman Arduino	27
Gambar 3.5 Perancangan <i>Prototype</i>	27
Gambar 4.1 Hasil Rancangan Keseluruhan	28
Gambar 4.2 Pengujian <i>Load Cell</i>	29
Gambar 4.3 Pengujian <i>Power Supply</i>	29
Gambar 4.4 Pengujian LCD.....	30
Gambar 4.5 <i>Regresi Linear</i>	35

DAFTAR TABEL

Tabel 2.1 Spesifikasi Arduino Uno.....	8
Tabel 4.1 Pengujian <i>Power Supply</i>	33
Tabel 4.2 Pengujian Kalibrasi <i>Load Cell</i> 5000 Gram.....	34
Tabel 4.3 Hasil Pengukuran <i>Load Cell</i>	35
Tabel 4.4 Hasil Pengukuran Timbangan dengan <i>Output</i> Suara.....	36

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pengukuran berat suatu buah sering menjadi hal penting bagi para penjual buah untuk memastikan harga yang sesuai dengan hasil timbangan yang diukur. Seiring dengan perkembangan teknologi dan ilmu pengetahuan, akhir ini bidang elektronika mengalami kemajuan yang pesat. Dengan kemajuan tersebut, membuat manusia selalu berusaha memanfaatkan teknologi yang ada untuk mempermudah kehidupan. Model timbangan buah dipasaran umumnya masih menggunakan timbangan tradisional, yang sering masih menghasilkan pengukuran yang tidak elit atau akurat. Selain itu alat yang ukuran yang lain juga hanya memakai timbangan analog yang *output* nya ditunjukkan oleh jarum penunjuk. Hasil pengukuran yang ditunjukkan oleh jarum penunjuk tidak menghasilkan pembacaan yang tepat. Hasil pembacaan dari jarum penunjuk masing-masing penjual masih memiliki pengukuran yang berbeda-beda. Jenis timbangan bermacam-macam, mulai dari timbangan manual, timbangan mekanik hingga timbangan digital.

Beberapa kendala yang dialami para penjual adalah menyetel ulang posisi jarum penunjuk supaya menjadi 0 baru kemudian mereka melakukan timbang menimbang.

Berdasarkan permasalahan tersebut, maka dirancang suatu alat yang mampu mengukur berat yang dilengkapi dengan hasil keluaran berupa suara agar kemungkinan kesalahan dalam proses pembacaan data dapat diperkecil serta mempermudah pemakaian dalam melihat pengukuran.

Pengembangan dilakukan dengan mengubah sensor yang dipakai dalam pengukuran benda yaitu menggunakan *load cell* yang mana alat ini mampu mengukur hingga berat maksimal mencapai 5kg. Selain itu media keluaran dari hasil pengukuran ini dapat dilihat dan di dengar melalui LCD dan speaker.

1.2. Rumusan Masalah

Untuk mengaplikasikan alat timbangan digital berbasis arduino uno dengan *output* suara yang harus dikerjakan adalah:

1. Bagaimana cara membuat tempat timbangan digital?
2. Bagaimana cara membuat rangkaian *load cell*?
3. Bagaimana cara membuat program kontrol?

1.3. Batasan Masalah

1. Timbangan buah *digital* ini digunakan untuk mengukur berat buah dengan berat maksimum 5kg dan suara hanya bekerja atau berjalan sampai 1kg.
2. *Load cell* sebagai sensor berat untuk mendeteksi berat buah.
3. Menggunakan *Df Player* sebagai menyimpan suara dan keluaran nya menggunakan speaker.

1.4. Tujuan

Adapun tujuan dari penelitian tugas akhir ini adalah:

1. Supaya para pembeli merasa tidak dicurangi oleh pedagang.
2. Mengetahui cara kerja sensor berat *load cell* untuk mengukur berat buah.

1.5. Manfaat Penelitian

Manfaat penelitian ini antara lain adalah:

1. Dengan adanya alat timbangan buah *digital* berbasis Arduino Uno ini para pembeli tidak perlu khawatir dengan masalah pengukuran yang tidak akurat lagi.
2. Dapat mempermudah pekerjaan para penjual karena timbangan ini memakai sistem Arduino.

1.6. Sistematik Laporan

Dalam penulisan tugas akhir ini, penulis membuat sistematika penulisan sebagai berikut:

BAB I PENDAHULUAN

Menguraikan secara singkat tentang latar belakang, perumusan masalah, batasan masalah, tujuan, manfaat dan sistematika penulisan laporan tugas akhir.

BAB II LANDASAN TEORI

Menguraikan secara singkat tentang jurnal-jurnal yang pernah ada dan landasan teori secara umum yang disertai dengan teori-teori dasar.

BAB III METODE PENELITIAN

Bab ini membahas tahap perancangan dan proses pembuatan tugas akhir.

BAB IV HASIL PENELITIAN DAN ANALISA

Bab ini membahas secara keseluruhan dari sistem dan dilakukan pengujian serta analisa pada setiap pengujian perangkat keras. Mengintegrasikan seluruh sistem dan pengujian, berdasarkan data dari hasil pengujian dan melakukan analisa terhadap keseluruhan sistem.

BAB V KESIMPULAN DAN SARAN

Berisikan kesimpulan terhadap hasil perancangan yang telah dibuat dan saran untuk perbaikan terhadap hasil tugas akhir yang telah dibuat.

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Marpaung dan kawan-kawan (2015), “Perancangan Sistem Pengontrolan Pengukuran Berat Pada Timbangan Kendaraan Secara Automatis”. Dalam hal pengukuran massa, pengukuran massa biasanya dilakukan secara manual dengan menggunakan timbangan manual. Timbangan merupakan alat bantu untuk mengetahui berat suatu benda. Salah satu penggunaan timbangan di bidang perindustrian dan untuk jalan umum, untuk mengetahui berat muatan suatu kendaraan. Pada tulisan ini, dirancang suatu alat pengontrolan berat muatan kendaraan dengan menggunakan mikrokontroler ATmega 8535 sebagai pengendali. Alat ini untuk menimbang berat muatan kendaraan secara otomatis. Alat ini menggunakan satu buah sensor yaitu load cell. Sensor diletakkan di tengah agar alat dapat menimbang secara baik. Pada saat alat mendeteksi adanya beban. Secara otomatis, sensor akan membaca dan mengirimkan sinyal ke mikrokontroler yang kemudian berat muatan kendaraan tersebut ditampilkan oleh LCD disertakan mengirim pesan lewat sms. Berdasarkan hasil analisis data yang dilakukan dengan skala 1 : 1000 yang artinya 1 kg tampilan di LCD sama dengan berat 1 ton yang sebenarnya. Pesan normal load akan terkirim apabila berat muatan kendaraan diantara $2000 \text{ kg} \leq 5000 \text{ kg}$. Apabila berat muatan kendaraan melebihi dari 5000 kg maka pesan over load akan terkirim.

Yandra dan kawan-kawan (2016), “Rancang Bangun Timbangan Digital Berbasis Sensor Beban 5 Kg Menggunakan Mikrokontroler Atmega328”. Telah dibangun sebuah prototipe alat untuk mengukur massa, gaya berat dan massa jenis menggunakan mikrokontroler ATmega328. Sistem pengukuran massa dan gaya berat berdasarkan prinsip kerja sensor beban 5 Kg. sedangkan untuk pengukuran massa jenis menggunakan prinsip kerja sensor beban dan sensor ultrasonik PING HC SR04. Pada pengukuran massa jenis benda digunakan prinsip perubahan volume air dalam tabung. Selanjutnya digunakan mikrokontroler ATmega328

sebagai pengolah data secara keseluruhan. Tahapan penelitian meliputi proses perancangan, pembuatan dan pengujian alat. Hasil eksperimen menunjukkan bahwa kinerja alat ini mampu mengukur massa 5 gram sampai dengan 4000 gram dengan error pengukuran sebesar 1,05%. Sedangkan error pengukuran massa jenis yang dibaca alat ini sebesar 1,64%. Hasil pembacaan dari alat memerlukan waktu sekitar lima detik untuk dijadikan acuan identifikasi data yang dihasilkan.

Aghili. F (2010), “Desain Load Cell Dengan Kapasitas Overload Yang Besar”. untuk meningkatkan rasio signal-to-noise (s / n) dan sensitivitas load cell, hal ini diinginkan rancang struktur yang menghasilkan regangan besar dekat dengan regangan maksimum sensor yang diizinkan bahan untuk kekuatan beban rating yang diberikan. namun, mengakomodasi margin keamanan dengan menghormati overloading, kompromi sensitivitas. makalah ini menyajikan desain, analisis, dan pengujian prototipe dari sel beban yang dapat memberikan kapasitas perlindungan beban berlebih yang besar tanpa mengorbankan sensitivitas sensor. ini dicapai dengan desain sensor khusus struktur yang menjadi hampir kaku setelah lenturnya mencapai defleksi maksimumnya, dengan demikian sensor dapat dilindungi terhadap beban berlebih yang besar. dimensi sensor, yang dimaksimalkan sensitivitas sensor, karena nilai muatan dan kelebihan muatan yang diberikan diperoleh melalui analisis kekuatan mekanik. prototipe sel beban dibuat dan kemudian diuji untuk mengukurnya karakteristik linearitas dan kelebihan. hasil eksperimen menunjukkan akurasi 0,2% penuh skala dan perlindungan berlebih dari lentur sensor.

2.2 Landasan Teori

Load cell adalah sebuah sensor berat untuk menimbang buah-buahan dan sayur-sayuran yang sering dipakai oleh para penjual. Alat ini akan bekerja jika diletakkan buah-buahan dan sayur-sayuran,

LCD adalah sebuah tampilan layar kecil yang berukuran 16x2 untuk menampilkan angka, nilai dari *load cell* yang biasa digunakan untuk para penjual. Alat ini akan bekerja apabila *load cell* diberi beban berupa buah-buahan dan sayur-sayuran. Selain itu juga, ada komponen lainnya seperti arduino uno, *mp3*

player, speaker dan alat-alat lainnya serta properti yang akan digunakan dalam pembuatan alat tersebut.

2.3 komponen-komponen yang digunakan

2.3.1 ArduinoUno

Arduino uno adalah *board* berbasis mikrokontroler pada ATmega328 *Board* ini memiliki 14 *digital input / output pin* (dimana 6 pin dapat digunakan Sebagai *output PWM*), 6 *input analog*, 16 MHz *osilator* kristal, koneksi USB, *jack* Listrik tombol *reset*. *Pin-pin* ini berisi semua yang diperlukan untuk mendukung mikrokontroler, hanya terhubung kekomputer dengan kabel USB atau sumber tegangan bisa didapat dari adaptor AC-DC atau baterai untuk menggunakannya.



Gambar 2.1 Arduino Uno
(Sumber: <https://ilearning.arduino-uno>, 2018)

Komponen utama pada arduino uno adalah sebuah mikrokontroler Atmega328 dengan konfigurasi *pin*.



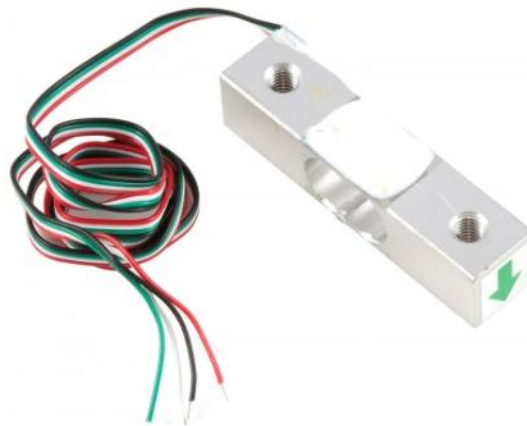
Gambar 2.2. Atmega 328 Pin Mapping
(Sumber: www.researchgate.net, 2018)

1. LED kaki 13, LED *indicator* catu daya

2. Kaki I/O *digital* (D8-D13)
3. Kaki I/O *digital* (D0-D7)
4. Tombol *reset*
5. Atmega328
6. Kaki I/O *Analog* (A0-A5)
7. Kaki *catu daya* (5V,GND)
8. Terminal *catu daya* (6-9V)

2.3.2 *Load Cell*

Load cell adalah komponen utama disistem timbangan, tingkat keakurasian dan kepresisian timbangan tergantung dari jenis *load cell* yang dipakai, sedangkan keakurasian timbangan akan menentukan efisiensi pada dunia industri, transaksi jual beli yang menggunakan satuan kilogram diwajibkan menggunakan timbangan yang dilegalisasi oleh departemen perdagangan melalui direktorat metrologi yang berwenang untuk melakukan TERA maksimal setahun sekali, dikarenakan semua timbangan dalam proses pemakaiannya pada jangka waktu tertentu akan mengalami deformasi mekanis pada *frame* timbangan.



Gambar 2.3 *Load Cell*
(Sumber :Sparkfun.com, 2018)

Dalam hal ini akan berpengaruh terhadap tingkat keakurasian dari *load cell* dan pada timbangan tersebut, sehingga harus dilaksanakan suatu sistem kalibrasi menggunakan *test weight* atau standar dari metrologi. Kalibrasi tersebut

adalah untuk menyamakan tegangan *load cell* dibandingkan dengan batu standar yang dibaca di *indicator* timbangan.

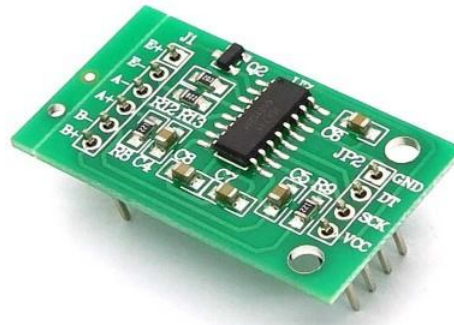
Diwajibkan untuk melakukan TERA maksimal setahun sekali, dikarenakan semua timbangan dalam proses pemakaiannya pada jangka waktu tertentu akan mengalami formasi mekanis pada *frame* timbangan. Dalam hal ini akan berpengaruh terhadap tingkat keakurasiannya dari *load cell* dan pada timbangan tersebut sehingga harus dilaksanakan kalibrasi menggunakan *test weight* atau batu standar dari metrologi. Kalibrasi tersebut adalah untuk menyamakan tegangan *load cell*.

Output load cell tidak hanya ditentukan oleh berat dan beban yang diterima tetapi juga oleh kekuatan tegangan *eksitasi*, yang dinilai dalam *output* sensitivitas mV/V pada kapasitas maksimal penuh. Sebuah keluaran khas beban maksimal untuk *load cell* adalah 3 *milivolt* atau *volt* (mV/V). Ini berarti bahwa untuk setiap *volt* tegangan *eksitasi* diterapkan pada beban maksimal akan ada 3 *milivolt* *output signal*.

2.3.3 IC HX711

IC HX711 adalah sebuah komponen terintegrasi dari perusahaan “AVIA SEMICONDUCTOR”, IC HX711 presisi 24 bit *analog to digital converter* (ADC) yang didesain untuk sensor timbangan digital (*weight scale*) dan industri control aplikasi yang terkoneksi dengan sensor jembatan (*bridge sensor*)

IC HX711 adalah model timbangan, yang memiliki prinsip kerja mengkonversi perubahan yang terukur dalam perubahan resistansi dan mengkonversinya kedalam besaran tegangan melalui rangkaian yang ada. Timbangan melakukan komunikasi dengan *computer* mikrokontroler melalui TTL232.



Gambar 2.4 IC HX711
(Sumber : www.itead.cc, 2018)

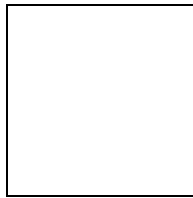
Adapun Prinsip kerja sensor regangan ketika mendapat tekanan beban. (sumber datasheet HX711) Ketika bagian lain yang lebih elastic mendapat tekanan, maka pada sisi lain akan mengalami perubahan regangan yang sesuai dengan yang dihasilkan oleh strain gauge, hal ini terjadi karena ada gaya yang seakan melawan pada sisi lainnya. Perubahan nilai resistansi yang diakibatkan oleh perubahan gaya diubah menjadi nilai tegangan oleh rangkaian pengukuran yang ada. Dan berat dari objek yang diukur dapat diketahui dengan mengukur besarnya nilai tegangan yang timbul.



Gambar Hubungan pin antara mikrokontroler dengan modul HX711
(Sumber: <http://www.indo-ware.com/blog-27-timbangan-5kg-hx711.html>)

2.3.4 LCD (*Liquid Cristal Display*)

LCD adalah lapisan dari campuran organik antara lapisan kaca bening dengan elektroda transparan indium oksida dalam bentuk tampilan *seven-segment* dan lapisan elektroda pada kaca belakang. Ketika elektroda diaktifkan dengan medan listrik (tegangan), molekul organik yang panjang dan silindris menyesuaikan diri dengan elektroda dari *segmen*. Lapisan *sandwich* memiliki *polarizer* cahaya *vertikal* depan dan *polarizer* cahaya *horizontal* belakang yang diikuti dengan lapisan *reflektor*. Cahaya yang dipantulkan tidak dapat melewati molekul-molekul yang telah menyesuaikan diri dan *segmen* yang diaktifkan terlihat menjadi gelap dan membentuk karakter data yang ingin ditampilkan.



Gambar 2.5 LCD

(Sumber :www.electronicwings.com,2018)

Dalam modul LCD (*Liquid Cristal Display*) terdapat mikrokontroler yang berfungsi sebagai pengendali tampilan karakter LCD (*Liquid Cristal Display*) mikrokontroler pada suatu LCD dilengkapi dengan memori dan *register*. Memori yang digunakan mikrokontroler *internal* LCD adalah :

1. DDRAM (*Display Data Random Access Memory*) merupakan tempat karakter yang akan ditampilkan berada.
2. CGRAM (*Character Generator Random Access Memory*) merupakan memori untuk menggambarkan pola sebuah karakter dimana bentuk dari karakter dapat diubah-ubah sesuai dengan keinginan.
3. CGROM (*Character Generator Read Only Memory*) merupakan memori untuk menggambarkan pola sebuah karakter dimana pola tersebut merupakan karakter dasar yang sudah ditentukan secara permanen oleh pabrikan pembuat LCD (*Liquid Cristal Display*) tersebut sehingga penggunaan tinggal mengambilnya sesuai alamat

memorinya dan tidak dapat merubah karakter dasar yang ada dalam CGROM.

Register control yang terdapat dalam suatu LCD diantaranya adalah :

1. *Register* perintah yaitu *register* yang berisi perintah-perintah dari mikrokontroler ke panel LCD (*liquid cristal display*) pada saat proses penulisan data atau tempat status dari panel LCD (*liquid cristal display*) dapat dibaca pada saat pembacaan data.
2. *Register* data yaitu *register* untuk menuliskan atau membaca data dari atau ke DDRAM. Penulisan data pada *register* akan menempatkan data tersebut ke DDRAM sesuai dengan alamat yang telah diatur sebelumnya.

Pin kaki atau jalur input dan kontrol dalam suatu LCD diantaranya adalah :

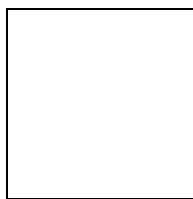
1. *Pin* data adalah jalur untuk memberikan data karakter yang ingin ditampilkan menggunakan LCD dapat dihubungkan dengan busa data dari rangkaian lain seperti mikrokontroler dengan lebar data 8 bit.
2. *Pin RS (register select)* berfungsi sebagai indikator atau yang menentukan jenis data yang masuk, apakah data atau perintah. Logika *low* menunjukkan yang masuk adalah perintah, sedangkan logika *high* menunjukkan data.
3. *Pin R/W (read write)* berfungsi sebagai instruksi pada modul jika *low* tulis data, sedangkan *high* baca data.
4. *Pin E (enable)* digunakan untuk memegang data baik masuk atau keluar.
5. *Pin VLCD* berfungsi mengatur kecerahan tampilan (kontras) dimana pin ini dihubungkan ke *ground*, sedangkan tegangan catu daya ke LCD sebesar 5 Volt.

2.3.5 Power Supply

Rangkaian *power supply* adalah suatu rangkaian yang digunakan untuk menyediakan sumber daya dalam bentuk keluaran tegangan. Tegangan yang dihasilkan disalurkan ke berbagai rangkaian lain untuk mengaktifkan rangkaian-rangkaian tersebut. Disini diperlukan rangkaian *power supply* DC +5 V dan DC +12 V yang digunakan untuk menyediakan *supply* sebesar DC +5 V dan DC +12 V. Untuk rangkaian *relay* dibutuhkan tegangan sebesar DC +12 V dari *power supply* dan DC +12 V dari aki untuk menggerakkan motor DC, sedangkan untuk rangkaian sistem minimum mikrokontroler, dan rangkaian sensor hanya membutuhkan tegangan sebesar DC +5 V.

2.3.6 Df Player

Df Player Mini adalah modul MP3 kompak dan dapat langsung dihubungkan ke *speaker*. Modul dengan baterai *power supply*, *speaker*, *keypad* dapat digunakan sendiri, juga dapat dikontrol melalui *port serial*, modul Arduino Uno untuk atau mikrokontroler seri. Modul itu sendiri sempurna terintegrasi *hardware decode MP3, WAV, WMA*. Sementara *driver* kartu TF dukungan perangkat lunak mendukung FAT16, sistem *file* FAT32. Dapat dilakukan dengan perintah serial sederhana. Mainkan musik, serta cara bermain musik dan fungsi lainnya, tanpa operasi yang mendasari rumit, mudah digunakan, stabil dan dapat diandalkan.



Gambar 2.6 Df Player

(Sumber: markus-wobisch.blogspot.com, 2018)

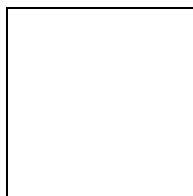
Aplikasi:

1. Siaran suara navigasi mobil.
2. Inspektur transportasi jalan, stasiun tol konfirmasi suara.

3. Stasiun kereta api, terminal bus pemeriksaan keamanan konfirmasi suara.
4. Listrik, komunikasi, ruang bisnis keuangan konfirmasi suara listrik.
5. Kendaraan masuk dan keluar dari saluran untuk memverifikasi konfirmasi suara.
6. Channel perbatasan konfirmasi suara.
7. Alarm suara multi-channel atau peralatan panduan operasi suara .
8. Mobil listrik tamasya pemberitahuan suara aman mengemudi.
9. Peralatan listrik kegagalan alarm.
10. Suara *alarm* kebakaran.
11. Peralatan siaran otomatis, siaran *reguler* .

2.3.7 Transformator

Transformator yang berfungsi menurunkan tegangan catu utama AC menjadi tegangan tertentu yang dibutuhkan oleh peralatan. *Input* tegangan yang *relative* tinggi, digunakanlah sebuah *transformator step-down* dengan rasio lilitan yang sesuai untuk mengkonversi tegangan tinggi ketegangan rendah.



Gambar 2.7 *Transformator*
(Sumber: Dokumentasi, 2018)

Transformator atau trafo ini bekerja berdasarkan prinsip induksi elektromagnet dan hanya dapat bekerja pada tegangan yang berarus bolak balik (AC). *Transformator* (trafo) memegang peranan yang sangat penting dalam pendistribusian tenaga listrik. *Transformator* menaikkan listrik yang berasal dari pembangkit listrik PLN hingga ratusan kilo Volt untuk di distribusikan, dan kemudian *transformator* lainnya menurunkan tegangan listrik tersebut ke tegangan yang diperlukan oleh setiap rumah tangga maupun perkantoran yang pada umumnya menggunakan tegangan AC 220Volt.

Transformator atau trafo terdiri dari:

- Lilitan (*coil*) primer
- Lilitan (*coil*) sekunder
- Inti besi (*core*)

Sebuah *transformator* yang sederhana pada dasarnya terdiri dari 2 lilitan atau kumparan kawat yang terisolasi yaitu kumparan primer dan kumparan sekunder. Pada kebanyakan *transformator*, kumparan kawat terisolasi ini dililitkan pada sebuah besi yang dinamakan dengan inti besi (*core*). Ketika kumparan primer dialiri arus AC (bolak-balik) maka akan menimbulkan medan magnet atau *fluks* magnetik disekitarnya. Kekuatan Medan magnet (densitas *fluks* magnet) tersebut dipengaruhi oleh besarnya arus listrik yang dialirinya. Semakin besar arus listriknya semakin besar pula medan magnetnya. Fluktuasi medan magnet yang terjadi di sekitar kumparan pertama (primer) akan menginduksi GGL (gaya gerak listrik) dalam kumparan kedua (sekunder) dan akan terjadi pelimpahan daya dari kumparan primer ke kumparan sekunder. Dengan demikian, terjadilah pengubahan taraf tegangan listrik baik dari tegangan rendah menjadi tegangan yang lebih tinggi maupun dari tegangan tinggi menjadi tegangan yang rendah.

Sedangkan inti besi pada *transformator* atau trafo pada umumnya adalah kumpulan lempengan-lempengan besi tipis yang terisolasi dan ditempel berlapis-lapis dengan kegunaanya untuk mempermudah jalannya *fluks* Magnet yang ditimbulkan oleh arus listrik kumparan serta untuk mengurangi suhu panas yang ditimbulkan.

Jenis – Jenis *Transformator* :

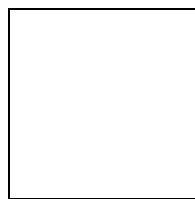
- Step Up*
- Step Down*
- Auto transformator*
- Auto transformator variabel*
- Transformator isolasi*
- Transformator pulsa*
- Transformator tiga fase*

2.3.8 *Speaker 5 Watt*

Speaker (bahasa Inggris) dalam bahasa Indonesia sering diistilahkan dengan "pengeras suara" adalah perangkat elektronik yang merubah getaran getaran listrik dalam spektrum audio menjadi getaran-getaran suara sehingga bisa terdengar oleh manusia. Dalam setiap sistem penghasil suara, penentuan kualitas suara terbaik tergantung dari *speaker*. Rekaman yang terbaik, dikodekan kedalam alat penyimpanan yang berkualitas tinggi, dan dimainkan dengan pengeras suara yang baik, tetap saja hasil suaranya akan jelek bila dihubungkan dengan *speaker* yang kualitasnya rendah.

Pada perancangan alat timbangan badan bersuara ini digunakan sebuah *mini speaker* yang terhubung ke *MP3 player* dimana tempat rekaman data suara disimpan. Pemilihan *mini speaker* ini disesuaikan pada bentuk desain sistem alat timbangan badan bersuara, sehingga akan terasa lebih pas bila dipergunakan sepasang *speaker mini*. Penggunaan *speaker mini* ini bertujuan membuat rancang bangun sistem lebih terlihat artistik.

Prinsip kerja dari *speaker* yaitu dalam menterjemahkan sinyal listrik menjadi suara yang dapat didengar, *speaker* memiliki komponen elektromagnetik yang terdiri dari kumparan yang disebut dengan *voice coil* untuk membangkitkan medan magnet dan berinteraksi dengan magnet permanen sehingga menggerakkan *cone speaker* maju dan mundur. *Voice coil* adalah bagian yang bergerak sedangkan magnet permanen adalah bagian *speaker* yang tetap pada posisinya. Sinyal listrik yang melewati *voice coil* akan menyebabkan arah medan magnet berubah secara cepat sehingga terjadi gerakan "tarik" dan "tolak" dengan magnet permanen. Dengan demikian, terjadilah getaran yang maju dan mundur pada *cone speaker*.



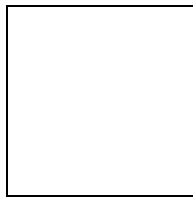
Gambar 2.8 *Speaker 5 Watt*
(Sumber: Diycade.com, 2018)

Cone adalah komponen utama *speaker* yang bergerak. Pada prinsipnya, semakin besarnya *cone* semakin besar pula permukaan yang dapat menggerakkan udara sehingga suara yang dihasilkan *speaker* juga akan semakin besar. *Suspension* yang terdapat dalam *speaker* berfungsi untuk menarik *cone* keposisi semula setelah bergerak maju dan mundur. *Suspension* juga berfungsi sebagai pemegang *cone* dan *voice coil*. Kekakuan (*rigidity*), komposisi dan desain *suspension* sangat mempengaruhi kualitas suara *speaker* itu sendiri.

2.3.9 Handphone

Handphone merupakan alat komunikasi dua arah yang bisa dibawa kemana-mana dan memiliki kemampuan untuk mengirimkan pesan berupa suara. Apalagi dengan semakin berkembangnya *handphone* sehingga *handphone* memiliki berbagai fungsi sekaligus. Bukan hanya sebagai alat komunikasi saja namun telah berkembang menjadi alat dengan fungsi lainnya seperti sebagai media hiburan, medis bisnis, dan sebagainya.

Fungsi utama dari *handphone* adalah sebagai alat komunikasi melalui suara dan pesan singkat (SMS). Selanjutnya *handphone* berfungsi untuk menangkap siaran radio, televisi. Juga dilengkapi dengan fungsi *audio*, kamera, video, *game*, serta layanan *internet*.

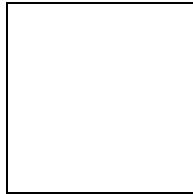


Gambar 2.9 *Handphone*
(Sumber: Dokumentasi, 2018)

2.3.10 Kabel Jumper

Kabel *jumper* adalah kabel penghubung yang biasa digunakan untuk membuat rangkaian sistem atau *prototype* sistem menggunakan arduino dan *breadboard*. Kabel *jumper* adalah komponen yang wajib ada saat belajar rangkaian elektronika dan komponen penghubung rangkaian arduino dengan *breadboard*.

Fungsi kabel *jumper* ini dapat digunakan untuk menyambungkan komponen elektronik yang satu dengan yang lainnya pada saat membuat proyek *prototipe* dengan menggunakan *breadboard*.

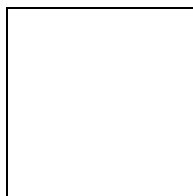


Gambar 2.10 Kabel *Jumper*
(Sumber :www.tokoarduino.com, 2018)

2.3.11 Resistor

Resistor adalah komponen elektronika yang berfungsi untuk menghambat atau membatasi aliran listrik yang mengalir dalam suatu rangkaian elektronika. Sebagaimana fungsi resistoryang sesuai namanya bersifat resistif dan termasuk salah satu komponen elektronika dalam kategori komponen pasif. Satuan atau nilai resistansi suatu resistor di sebut *ohm* dan dilambangkan dengan simbol *Omega* (Ω). Sesuai hukum *ohm* bahwa resistansi berbanding terbalik dengan jumlah arus yang mengalir melaluinya. Selain nilai resistansinya (*Ohm*) resistorjuga memiliki nilai yang lain seperti nilai toleransi dan kapasitas daya yang mampu dilewatkannya. Semua nilai yang berkaitan dengan resistor tersebut penting untuk diketahui dalam perancangan suatu rangkaian elektronika oleh karena itu pabrikan resistor selalu mencantumkan dalam kemasan resistor tersebut.

Berikut adalah simbol resistor dalam bentuk gambar yang sering digunakan dalam suatu desain rangkaian elektronika.



Gambar 2.11 Simbol Resistor
(Sumber: Zonaelektro.net, 2018)

Resistor dalam suatu teori dan penulisan formula yang berhubungan dengan resistor disimbolkan dengan huruf “R”. Kemudian pada desain skema elektronika resistor tetap disimbolkan dengan huruf “R”, resistor variable disimbolkan dengan huruf “VR” dan untuk resistor jenis potensiometer ada yang disimbolkan dengan huruf “VR” dan “POT”.

Jenis-Jenis Resistor

1. Resistor Tetap (*fixed resistor*)

Resistor tetap merupakan resistor yang nilai resistansinya tidak dapat diubah atau tetap. Resistor jenis ini biasa digunakan dalam rangkaian elektronika sebagai pembatas arus dalam suatu rangkaian elektronika. Resistor tetap dapat kita temui dalam beberapa jenis, seperti :

1. *Metal film resistor*
2. *Metal oxide resistor*
3. *Carbon film resistor*
4. *Ceramic encased wirewound*
5. *Economy wirewound*
6. *Zero ohm jumper wire*
7. *SIP resistor network*

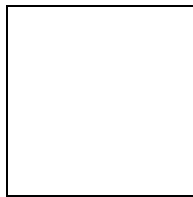
2. Resistor Tidak Tetap (*variable resistor*)

Resistor tidak tetap atau *variable resistor* terdiri dari 2 tipe yaitu :

1. *Potensiometer*, tipe *variable resistor* yang dapat diatur nilai resistansinya secara langsung karena telah dilengkapi dengan tuas kontrol. *Potensiometer* terdiri dari 2 jenis yaitu *potensiometer linier* dan *potensiometer logaritmis*
2. *Trimer potensiometer*, yaitu tipe *variable resistor* yang membutuhkan alat bantu (obeng) dalam mengatur nilai resistansinya. Pada umumnya resistor jenis ini disebut dengan istilah “*trimer potensiometer* atau VR”
3. *Thermistor*, yaitu tipe resistor *variable* yang nilai resistansinya akan berubah mengikuti suhu disekitar resistor. *Thermistor* terdiri dari 2 jenis yaitu NTC dan PTC.

2.3.12 Kapasitor

Kapasitor adalah salah satu jenis komponen elektronika yang memiliki kemampuan dapat menyimpan muatan arus listrik di dalam medan listrik selama batas waktu tertentu dengan cara mengumpulkan ketidakseimbangan *internal* dari muatan arus listrik tersebut. Kapasitor juga memiliki sebutan lain, yakni kondensator. Kapasitor atau kondensator ini termasuk salah satu jenis komponen pasif. Komponen yang satu ini ditemukan pertama kali oleh seorang ilmuwan bernama Michael Faraday yang lahir pada tahun 1791, dan wafat pada 1867. Karena itu satuan yang digunakan untuk kapasitor adalah Farad (F) yang diambil dari nama ilmuwan tersebut. Bentuk fisik kapasitor dapat dilihat pada Gambar 2.12.



Gambar 2.12 Kapasitor
(Sumber : Belajarelektronika, 2018)

Fungsi dari kapasitor adalah sebagai berikut:

1. Untuk menyimpan arus dan tegangan listrik sementara waktu.
2. Sebagai penyaring atau *filter* dalam sebuah rangkaian elektronika seperti *power supply* atau adaptor.
3. Untuk menghilangkan *bouncing* (percikan api) apabila dipasang pada saklar.
4. Sebagai isolator atau penahan arus listrik untuk arus DC atau searah.
5. Untuk meratakan gelombang tegangan DC pada rangkaian pengubah tegangan AC ke DC (adaptor).
6. Sebagai *oscilator* atau pembangkit gelombang AC (bolak-balik).

Zat dielektrik yang digunakan untuk menyekat kedua komponen tersebut berguna untuk membedakan jenis-jenis kapasitor. Terdapat beberapa kapasitor

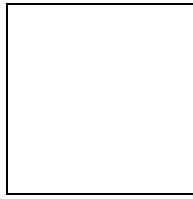
yang menggunakan bahan dielektrik, antara lain kertas, mika, plastik cairan dan masih banyak lagi bahan dielektrik lainnya. Dalam rangkaian elektronika kapasitor sangat diperlukan terutama untuk mencegah loncatan bunga api listrik pada rangkaian yang mengandung kumparan. Selain itu, kapasitor juga dapat menyimpan muatan atau energi listrik dalam rangkaian, dapat memilih panjang gelombang pada radio penerima dan sebagai *filter* alam catu daya (*power supply*).

2.3.13 Dioda

Dioda (*diode*) adalah komponen elektronika aktif yang terbuat dari bahan semikonduktor dan mempunyai fungsi untuk menghantarkan arus listrik ke satu arah tetapi menghambat arus listrik dari arah sebaliknya. Oleh karena itu, dioda sering dipergunakan sebagai penyearah dalam rangkaian elektronika. Dioda pada umumnya mempunyai 2 *elektroda* (terminal) yaitu anoda (+) dan katoda (-) dan memiliki prinsip kerja yang berdasarkan teknologi pertemuan P-N semikonduktor yaitu dapat mengalirkan arus dari sisi tipe-P (anoda) menuju ke sisi tipe-N (katoda) tetapi tidak dapat mengalirkan arus ke arah sebaliknya.

Berdasarkan fungsi dioda, dioda dapat dibagi menjadi beberapa jenis, diantaranya adalah:

1. Dioda penyearah (dioda biasa atau dioda *bridge*) yang berfungsi sebagai penyearah arus AC ke arus DC.
2. Dioda zener yang berfungsi sebagai pengaman rangkaian dan juga sebagai penstabil tegangan.
3. Dioda LED yang berfungsi sebagai lampu indikator ataupun lampu penerangan
4. Dioda *schottky* yang berfungsi sebagai pengendali.



Gambar 2.14 Dioda
(Sumber : Teknikelektronika.com, 2018)

Adapun jenis-jenis dan fungsi dioda adalah sebagai berikut :

1. Dioda *bridge* sebagai penyearah.
2. Dioda *zener* sebagai penstabil tegangan (*voltage regulator*).
3. Dioda sebagai pengaman atau *sekring*.
4. Dioda sebagai rangkaian *clipper*, yaitu untuk memangkas atau membuang level sinyal yang ada di atas atau di bawah level tegangan tertentu.
5. Dioda sebagai rangkaian *clamper*, yaitu untuk menambahkan komponen DC kepada suatu sinyal AC.
6. Dioda sebagai pengganda tegangan.
7. Dioda sebagai indikator untuk LED (*light emitting diode*).
8. Dioda sebagai sensor panas, contoh aplikasi pada rangkaian *power amplifier*.
9. Dioda foto sebagai sensor cahaya.
10. Dioda *varactor* sebagai rangkaian VCO (*voltage controlled oscillator*).

BAB III

METODE PENELITIAN

3.1 Tinjauan Umum

Timbangan buah dipasaran umumnya masih menggunakan timbangan tradisional (*konvensional*) yang sering kali masih menghasilkan pengukuran yang tidak teliti dikarenakan tidak adanya akurasi dan tingkat presisi. Dengan memanfaatkan sensor *load cell* sebagai pendeteksi objek berupa buah-buahan dan mengirimkan sinyal ke HX711 sebagai penguat keluaran dari *load cell* dan diteruskan ke arduino untuk mengirimkan instruksi ke LCD dan suara untuk mengaktifkan berapa berat buah tersebut.

Sistem ini sangat sulit dilakukan ditengah proses timbang-menimbang secara manual. Yaitu menginstal dengan cara memutar tombol *digital*. Dengan demikian, sistem ini akan membuat fungsi timbangan buah *digital* bekerja secara otomatis. Yaitu, tanpa menekan tombol *restart* atau menyetel ulang.

Untuk mempermudah dalam pemuatan tugas akhir ini, maka dibuat langkah-langkah perancangan alat yang akan dibuat. Adapun langkah-langkah dalam perancangan alat secara umum terbagi menjadi tiga bagian yaitu:

1. Desain *Prototype*

Pada tahap inidilakukan perancangan mengenai *prototype* yang akan dibangun, seperti model, tata letak komponen dan lain-lain.

2. Perangkat keras (*Hardware*)

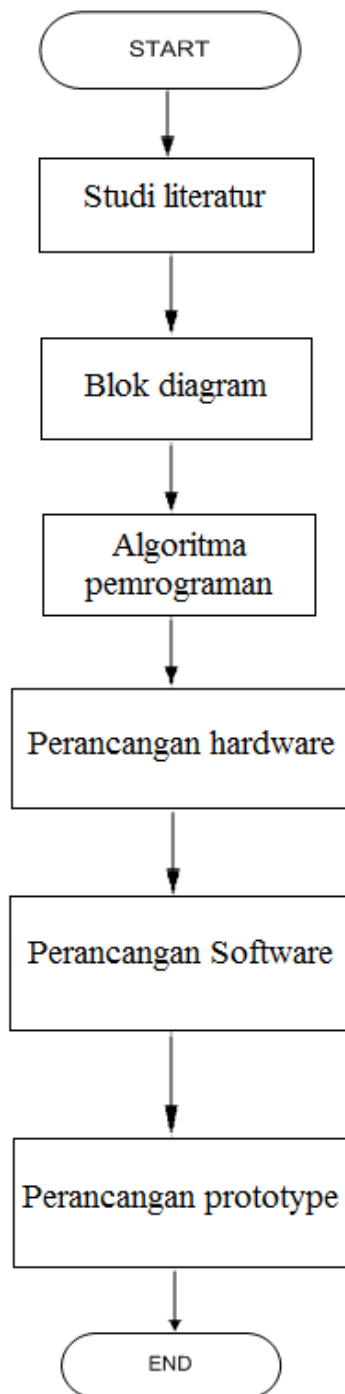
Pada tahap ini dilakukan pengujian dari tiap komponen apakah komponen layak dipakai atau tidak dan menguji pembacaan *load cell*.

3. Perangkat lunak (*Software*)

Pada tahap ini dilakukan pengisian program pada arduino uno, program tersebut yang akan mengatur seluruh proses kerja alat.

3.2 Metode Penelitian

Dalam penyelesaian penelitian tugas akhir ini dijadikan dalam bentuk metodologi penelitian seperti. Gambar 3.1 menunjukkan gambaran umum metodologi penelitian.



Gambar 3.1 Metodologi penelitian
(Sumber: *Dokumentasi*, 2018)

Berdasarkan Gambar 3.1 dapat dilihat bahwa penelitian dilakukan secara berurutan sehingga mencapai hasil akhir penulisan laporan pada pengambilan data, analisa dan kesimpulan dari seluruh pengujian pada alat.

3.3 Studi Literatur

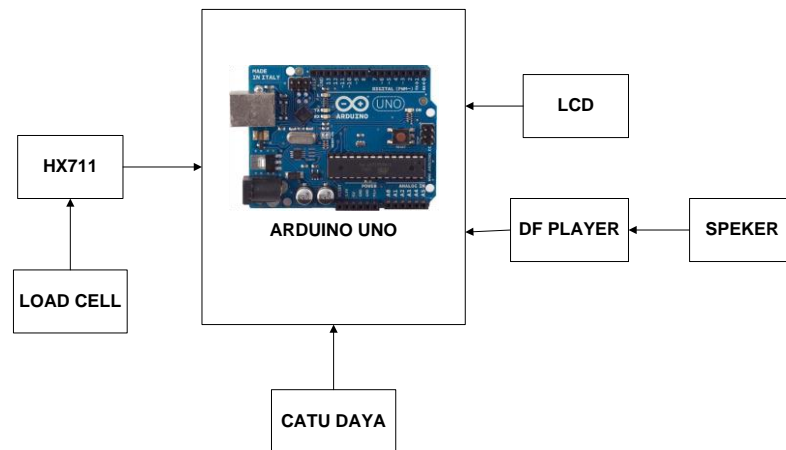
Studi literatur adalah mencari referensi teori yang relevan dengan kasus atau permasalahan yang ditemukan. Referensi tersebut berisikan tentang:

1. Dasar teori tentang *load cell*.
2. Dasar teori tentang HX711.
3. Dasar teori tentang *df player*.
4. Dasar teori tentang LCD
5. Sistem pengontrolan pada mikrokontroler untuk mengontrol timbangan otomatis.

Studi literatur adalah kumpulan dari seluruh referensi-referensi penelitian yang terdahulu dengan perumusan masalah yang ada. Referensi ini bisa dapat dicari dari buku, jurnal, artikel laporan penelitian dan situs-situs di *internet*. Tujuannya adalah untuk mempermudah dan memperkuat permasalahan serta sebagai landasan teori dalam melakukan studi dan juga dalam penyelesaian tugas akhir ini.

3.4 Blok Diagram

Blok diagram adalah suatu pernyataan gambar yang ringkas dari gabungan antara masukan, keluaran dan proses dari suatu sistem. Pada alat timbangan otomatis ini menggunakan *load cell* sebagai untuk menentukan suatu berat, HX711 sebagai penguat keluaran *load cell* agar lebih akurat, *df player* sebagai keluaran suara dan LCD sebagai keluaran untuk menampilkan berat buah. Gambar 3.2 menunjukkan gambaran umum sistem dalam bentuk blok diagram.



Gambar 3.2 Diagram Blok
(Sumber : Dokumentasi, 2018)

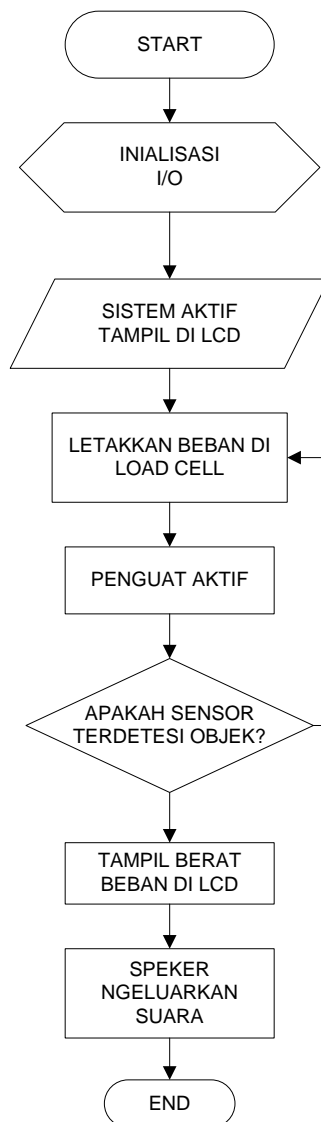
Gambar 3.2 dapat dilihat dari masing-masing komponen memiliki fungsi yang berbeda, adapun fungsinya sebagai berikut :

1. Load cell yang merupakan sensor transduser berat dapat merubah tekanan (*force*) menjadi sinyal listrik. Terdapat tiga cara load cell mengubah nilai tekanan (*force*) yang diterima menjadi nilai pengukuran, yaitu *hidraulic* load cell, *pneumatic* load cell dan *strain gauge* load cell. Pada sistem timbangan digital ini digunakan *strain gauge* load cell. Keluaran dari sensor ini resistansi yang merepresikan berat benda.
2. Modul HX711 adalah modul yang memudahkan kita membaca load cell dalam pengukuran berat. Modul ini berfungsi untuk menguatkan sinyal keluaran dari sensor dan mengonversi data analog menjadi data digital. Dengan menghubungkannya ke mikrokontroler, kita dapat membaca perubahan resistansi dari load cell. Setelah proses kalibrasi kita akan memperoleh pengukuran berat dengan keakuratan yang tinggi.
3. Arduino uno sebagai pengolah data akan membaca data output dari modul HX711. Karena output dari modul HX711 sudah dalam bentuk gram ini menyebabkan beban kerja arduino cukup ringan. Hasil pembacaan tadi selanjutnya dibulatkan dengan ketelitian 0,5 dan dikirim ke labview.
4. Labview akan menerima data dari arduino dan akan menampilkan data tersebut pada monitor PC.

3.5 Algoritma Pemrograman

Algoritma merupakan perangkat lunak dari sebuah *project* untuk menjalankan suatu perintah. Algoritma diterjemahkan ke bahasa pemrograman dan menjadi kesatuan yang disebut kecerdasan buatan. Algoritma dapat dijelaskan dalam bentuk *flowchart*.

Flowchart adalah suatu bagian dengan simbol-simbol tertentu yang menggambarkan urutan proses secara mendetail dan hubungan antara suatu proses (intruksi) dengan proses lainnya dalam suatu program.



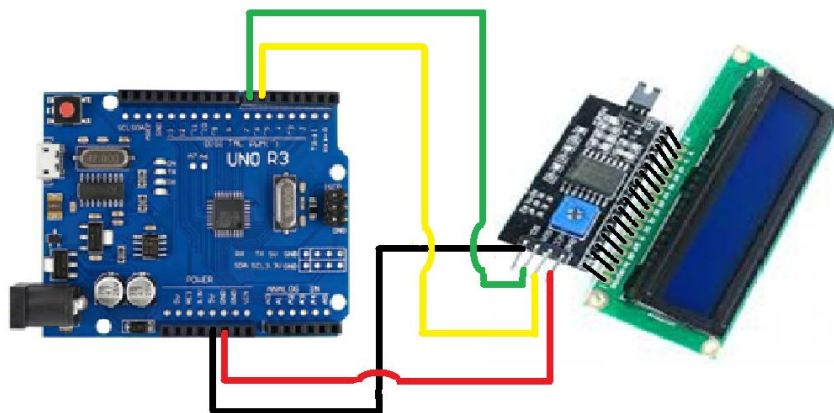
Gambar 3.3 Gambar *Flowchart*
(Sumber : Dokumentasi, 2018)

1. *Start* menandakan program siap.
2. Inisialisasi I/O mengenalkan rangkaian *input/output*.
3. Sistem aktif siap di jalankan dan tertampil di LCD.
4. Letakkan beban di *load cell* adalah untuk mendeteksi berat buah.
5. Penguat aktif adalah untuk memperkuat hasil timbangan menjadi akurat.
6. Apakah sensor terdeteksi objek adalah untuk mengetahui ada atau tidak nya suatu beban, jika iya berat akan tampil di LCD dan speaker akan mengeluarkan suara sesuai yang tertampil di LCD. Jika tidak letakkan kembali beban di *load cell*.

3.6 Perancangan *Hardware*

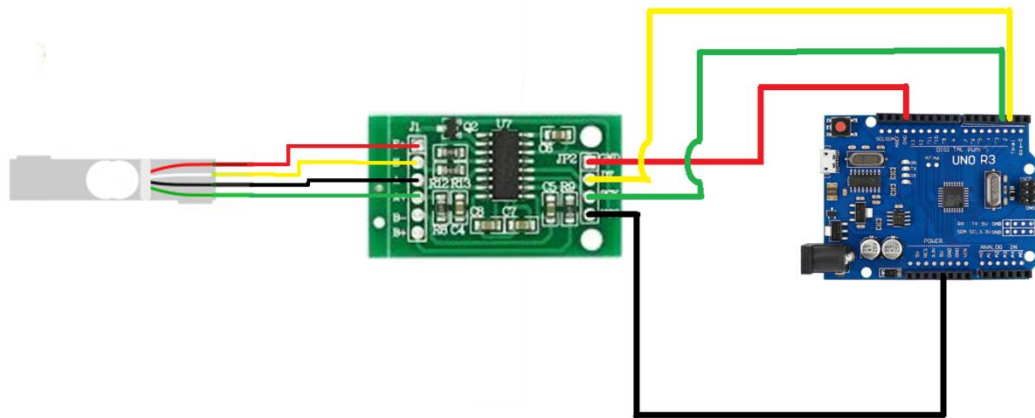
Pada penelitian ini menggunakan dua perancangan yaitu perancangan perangkat keras (*hardware*) dan perancangan lunak (*software*). Perancangan ini membuat sebuah alat timbangan buah *digital* yang berfungsi untuk memudahkan para penjual untuk menjual dan juga untuk menghasilkan pengukuran yang akurasi dan tingkat presisi tinggi, dengan cara mendeteksi ada atau tidaknya buah. Perancangan sebuah alat meliputi diagram blok rangkaian dan realisasi rangkaian dengan prinsip kerja dari masing-masing blok rangkaian, begitu juga yang digunakan pada perancangan timbangan buah digital berbasis Arduino Uno.

1. Koneksi LCD ke arduino



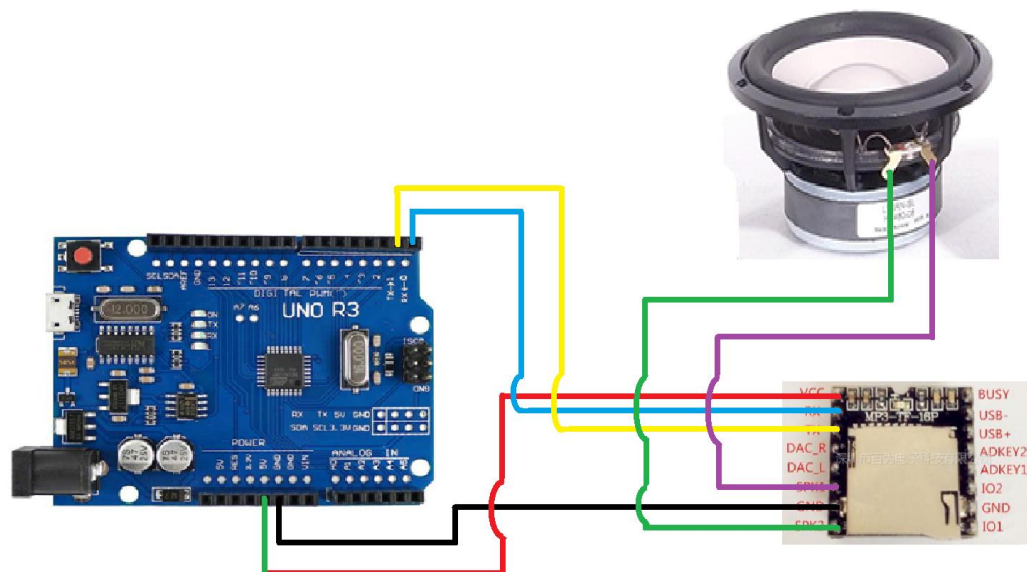
Gambar 3.4 koneksi LCD ke arduino
(Sumber : Dokumentasi, 2018)

2. Koneksi load cell ke HX711 dan arduino



Gambar 3.5 koneksi load cell ke HX711 dan arduino
(Sumber: Dokumentasi, 2018)

3. Koneksi speaker ke *df player* dan arduino



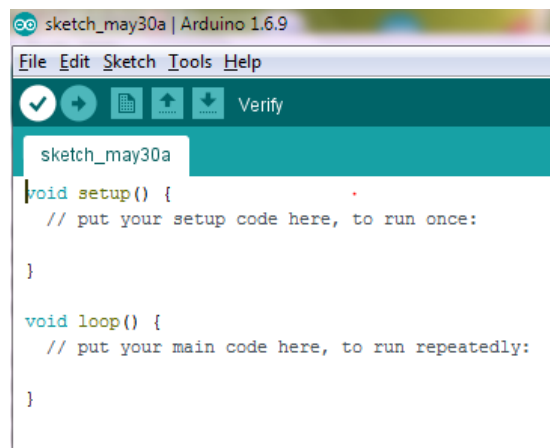
Gambar 3.6 koneksi speaker ke *df player* dan arduino
(Sumber: Dokumentasi, 2018)

3.7 Perancangan Software

3.7.1 Pemrograman Arduino

Arduino Uno dapat diprogram dengan menggunakan *software* arduino. *Software* ini bisa didapatkan secara gratis dari *website* resmi arduino. *Software* arduino yang akan digunakan adalah *driver* dan IDE. IDE arduino adalah *software* yang sangat canggih ditulis dengan menggunakan java IDE arduino terdiri dari :

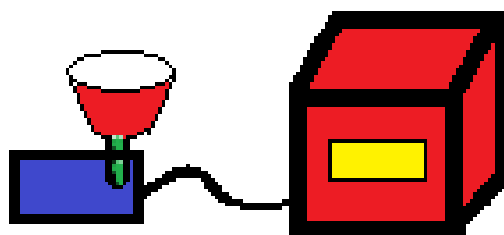
1. *Editor program*, sebuah *window* yang memungkinkan pengguna menulis dan mengedit program dalam bahasa *processing*.
2. *Compiler*, sebuah modul yang mengubah kode *program* (bahasa *processing*)menjadi kode biner. Bagaimanapun sebuah mikrokontroler tidak akan memahami bahasa *processing*, yang bisa dipahami oleh mikrokontroler adalah kode biner. Itula sebabnya *compiler* diperlukan dalam hal ini.
3. *Uploader*, sebuah modul yang memuat kode biner dari *computer* kedalam memori di dalam papan arduino.



Gambar 3.7 Arduino Uno
(Sumber : Dokumentasi, 2018)

3.8 Perancang *Prototype*

Rancang *prototype* alat timbangan *digital* berbasis arduino uno dengan *output* suara dapat dilihat dari gambar 3.8 dibawah ini :

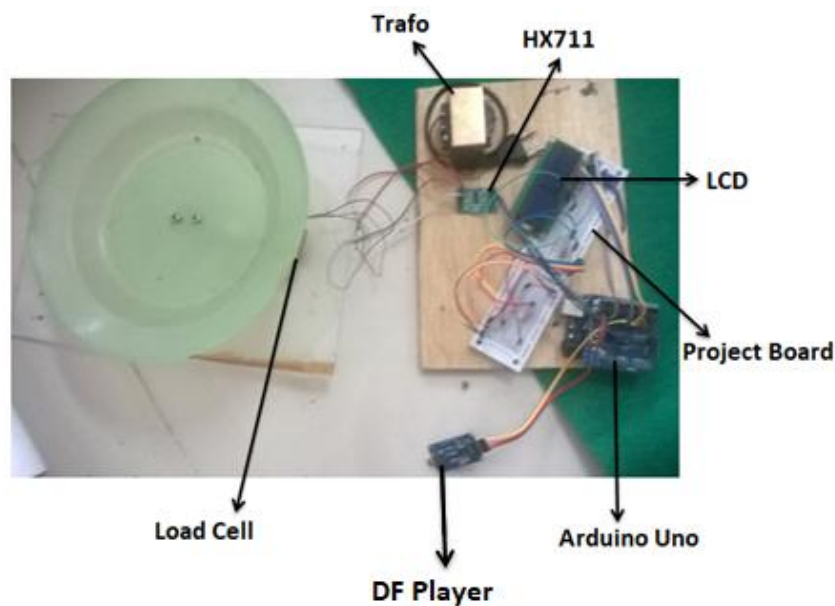


Gambar 3.8 Gambar Tampak Atas
(Sumber : Dokumentasi, 2018)

BAB IV HASIL PENELITIAN DAN ANALISA

4.1 Hasil Perancangan

Perancangan merupakan suatu proses permulaan sebelum melakukan suatu pekerjaan. Setelah pembuatan perancangan alat, maka langkah selanjutnya adalah pengujian alat dan analisa data dari alat *real* yang menggunakan *load cell*, *Df player* bekerja sesuai perancangan alat. Adapun pengujian yang dilakukan yaitu pengujian terhadap *hardware* maupun pengujian *software* serta analisa pembahasan terhadap data yang diperoleh. Gambar hasil perancangan dapat dilihat pada Gambar 4.1 di bawah ini:



Gambar 4.1 Hasil Rancangan Keseluruhan
(Sumber: Dokumentasi, 2018)

4.2 Pengujian *Load Cell*

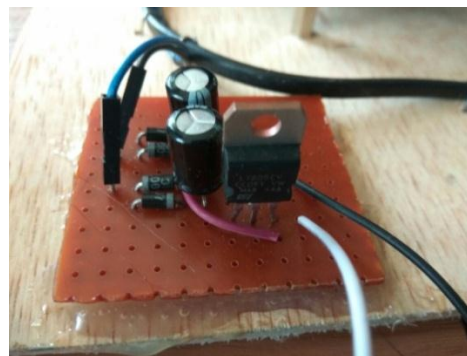
Pengujian pada *load cell* dengan meletakkan beban di atasnya, hasil pengujian dengan beban *maximum* 5kg yang diperoleh cepat dengan arduino akan memerintahkan LCD untuk menampilkan hasil berat. Seperti yang terlihat pada Gambar 4.2.



Gambar 4.2 Pengujian Load Cell
(Sumber: Dokumentasi Priadi, 2018)

4.3 Pengujian Rangkaian *Power Supply*

Power supply merupakan alat atau perangkat keras yang mampu men-*supply* tenaga atau tegangan listrik secara langsung dari sumber tegangan listrik ke tegangan listrik yang lainnya. Gambar 4.2. berikut adalah perancangan *power supply* yang telah dilaksanakan:



Gambar 4.3 *Power Supply* 5v
(Sumber: Dokumentasi Pribadi, 2018)

Tabel 4.1 berikut adalah hasil pengujian *power supply* 5VDC

No	Tegangan pada trafo	Vin AC Pengukuran	Vout DC Pengukuran
1.	0-4.5Volt	5.1 Volt	2.8 Volt
2.	0-6Volt	7.1 Volt	4.2 Volt
3.	0-7.5 Volt	8.8 Volt	5.4 Volt

(Sumber: Dokumentasi, 2018)

4.4 Pengujian LCD

Pengujian LCD (*Liquid Crystal Display*) ini berfungsi untuk mengetahui apakah LCD bisa bercahaya dan menampilkan tulisan dengan baik. *Liquid Crystal Display* pada alat ini digunakan untuk menampilkan suhu dan pwm kecepatan putaran kipas. Gambar 4.4 adalah pengujian dari *Liquid Crystal Display*:



Gambar 4.4 LCD
(Sumber: Dokumentasi Pribadi, 2018)

Dalam pengujian ini untuk mengaktifkan *Liquid Crystal Display* digunakan I2C agar lebih menghemat pin pada arduino. Pengujian ini juga dilengkapi dengan program supaya bisa menampilkan tulisan pada layar LCD.

4.5 Pengujian Alat Keseluruhan

Pengujian keseluruhan merupakan langkah yang digunakan untuk mengetahui sejauh mana kesesuaian antara rancangan dengan kenyataan pada alat yang telah dibuat, apakah sudah sesuai dengan yang diharapkan atau tidak.

Sesuai yang telah disebutkan pengujian keseluruhan ini antara lain:

1. Mengetahui tegangan power supply baik tegangan AC maupun DC.
2. Mengetahui apakah *load cell* mendeteksi ada beban dan tidak ada beban.
3. Mengetahui apakah *Df Player* bisa mengeluarkan suara sesuai dengan keluaran dari *load cell*.
4. Mengetahui apakah *Liquid Crystal Display* bisa menampilkan tulisan.

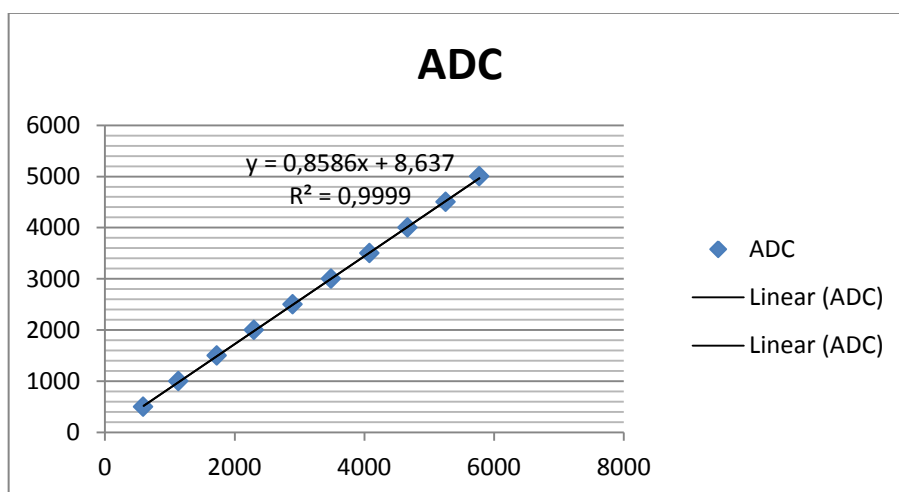
Tabel 4.2 Hasil kalibrasi pada *load cell*

NO	Input	Output Load Cell (X)	Hasil Kalibrasi Load Cell ($0,8586 * X + 8,637$)
1	500	594	518,64
2	1000	1138	985,72
3	1500	1730	1494,01
4	2000	2302	1985,13
5	2500	2896	2495,14
6	3000	3488	3003,43
7	3500	4082	3513,44
8	4000	4670	4018,29
9	4500	5256	4521,43
10	5000	5774	4966,19

(Sumber: Dokumentasi, 2018)

Pengujian kalibrasi berat menggunakan pasir sebanyak 5000gram, pasir yang sudah disiapkan dengan takaran 5000gram. *Output* dari *load cell* atau dinamakan dengan *variabel* (X) adalah satuan dari gram. Dikarenakan tipe *load cell* yang digunakan yaitu CZL 635, data yang didapatkan bisa dilihat pada tabel 4.2.

Grafik regresi linear untuk mendapatkan persamaan.



Gambar 4.5 Hasil Regresi Linier
(Sumber: Dokumentasi, 2018)

Dari hasil grafik didapatkan, maka persamaan regresi liniernya yaitu $y = 0.8586 * X + 8,637$ akan menjadi rumus persamaan yang dimasukkan didalam *coding* program arduino uno. *Variabel (X)* adalah hasil dari data kalibrasi yang ada pada sensor *load cell*.

Tabel 4.3 Hasil pengukuran timbangan buah

Nama Buah	Berat Asli (Gram)	Berat Pengukuran (Gram)	Selisih	Eror
Kedondong	75	84	8	9,5
Kentang	64	87	23	26,4
Tomat	55	75	20	26,6
$nilai\ rata - rata\ error = \frac{9,5 + 26,4 + 26,6}{3} = 20,8\%$				$\frac{20,8}{100} = 0,208\%$

(Sumber: Dokumentasi, 2018)

Untuk mendapatkan nilai selisih, persamaan yang digunakan:

$$Selisih = Hasil\ Kalibrasi - Input$$

Untuk mendapatkan nilai eror, persamaan yang dipakai:

$$Error = \frac{selisih}{hasil\ kalibrasi} \times 100$$

Tabel 4.4 Hasil pengukuran timbangan buah dengan output suara

Nama Buah	Tampilan LCD	Suara
Kedondong	84	Aktif
Kentang	87	Aktif
Tomat	75	Aktif

(Sumber : Dokumentasi, 2018)

Tabel 4.5 Hasil range suara

No	Nilai Gram	Range Gram
1	30 Gram	30 – 34,99 Gram
2	35 Gram	35 – 39,99 Gram
3	40 Gram	40 – 44,99 Gram
4	45 Gram	45 – 49,99 Gram
5	50 Gram	50 – 54,99 Gram
6	55 Gram	55 – 59,99 Gram
7	60 Gram	60 – 64,99 Gram
8	65 Gram	65 – 69,99 Gram
9	70 Gram	70 – 74,99 Gram
10	75 Gram	75 – 79,99 Gram
11	80 Gram	80 – 84,99 Gram
12	85 Gram	85 – 89,99 Gram
13	90 Gram	90 – 94,99 Gram
14	95 Gram	95 – 99,99 Gram
15	100 Gram	100 – 104,99 Gram
16	105 Gram	105 – 109,99 Gram
17	110 Gram	110 – 114,99 Gram
18	115 Gram	115 – 119,99 Gram
19	120 Gram	120 – 124,99 Gram
20	125 Gram	125 – 129,99 Gram
21	130 Gram	130 – 134,99 Gram
22	135 Gram	135 – 139,99 Gram
23	140 Gram	140 – 144,99 Gram
24	145 Gram	145 – 149,99 Gram
25	150 Gram	150 – 154,99 Gram
26	155 Gram	155 – 159,99 Gram
27	160 Gram	160 – 164,99 Gram
28	165 Gram	165 – 169,99 Gram
29	170 Gram	170 – 174,99 Gram
30	175 Gram	175 – 179,99 Gram
31	180 Gram	180 – 184,99 Gram
32	185 Gram	185 – 189,99 Gram
33	190 Gram	190 – 194,99 Gram
34	195 Gram	195 – 199,99 Gram
35	200 Gram	200 – 204,99 Gram
36	205 Gram	205 – 209,99 Gram
37	210 Gram	210 – 214,99 Gram
38	215 Gram	215 – 219,99 Gram
39	220 Gram	220 – 224,99 Gram
40	225 Gram	225 – 229,99 Gram
41	230 Gram	230 – 234,99 Gram
42	235 Gram	235 – 239,99 Gram
43	240 Gram	240 – 245,99 Gram
44	245 Gram	245 – 249,99 Gram
45	250 Gram	250 – 254,99 Gram
46	255 Gram	255 – 259,99 Gram
47	230 Gram	230 – 234,99 Gram
48	235 Gram	235 – 239,99 Gram
49	240 Gram	240 – 244,99 Gram
50	245 Gram	245 – 249,99 Gram

51	250 Gram	250 – 254,99 Gram
52	255 Gram	255 – 259,99 Gram
53	260 Gram	260 – 264,99 Gram
54	265 Gram	265 – 269,99 Gram
55	270 Gram	270 – 274,99 Gram
56	275 Gram	275 – 279,99 Gram
57	280 Gram	280 – 284,99 Gram
59	285 Gram	285 – 289,99 Gram
60	290 Gram	290 – 294,99 Gram
61	295 Gram	295 – 299,99 Gram
62	300 Gram	300 – 304,99 Gram
63	305 Gram	305 – 309,99 Gram
64	310 Gram	310 – 314,99 Gram
65	315 Gram	315 – 319,99 Gram
66	320 Gram	320 – 324,99 Gram
67	325 Gram	325 – 329,99 Gram
68	330 Gram	330 – 334,99 Gram
69	335 Gram	335 – 339,99 Gram
70	340 Gram	340 – 344,99 Gram
71	345 Gram	345 – 349,99 Gram
72	350 Gram	350 – 354,99 Gram
73	355 Gram	355 – 359,99 Gram
74	360 Gram	360 – 364,99 Gram
75	365 Gram	365 – 369,99 Gram
76	370 Gram	370 – 374,99 Gram
77	375 Gram	375 – 379,99 Gram
78	380 Gram	380 – 384,99 Gram
79	385 Gram	385 – 389,99 Gram
80	390 Gram	390 – 394,44 Gram
81	395 Gram	395 – 399,99 Gram
82	400 Gram	400 – 404,99 Gram
83	405 Gram	405 – 409,99 Gram
84	410 Gram	410 – 414,99 Gram
85	415 Gram	415 – 419,99 Gram
86	420 Gram	420 – 424,99 Gram
87	425 Gram	425 – 429,99 Gram
88	430 Gram	430 – 434,99 Gram
89	435 Gram	435 – 439,99 Gram
90	440 Gram	440 – 444,99 Gram
91	445 Gram	445 – 449,99 Gram
92	450 Gram	450 – 454,99 Gram
93	455 Gram	455 – 459,99 Gram
94	460 Gram	460 – 464,99 Gram
95	465 Gram	465 – 469,99 Gram
96	470 Gram	470 – 474,99 Gram
97	475 Gram	475 – 479,99 Gram
98	480 Gram	480 – 484,99 Gram
99	485 Gram	485 – 489,99 Gram
100	490 Gram	490 – 494,99 Gram
101	495 Gram	495 – 499,99 Gram
102	500 Gram	500 – 504,99 Gram
103	505 Gram	505 – 509,99 Gram

104	510 Gram	510 – 514,99 Gram
105	515 Gram	515 – 519,99 Gram
106	520 Gram	520 – 524,99 Gram
107	525 Gram	525 – 529,99 Gram
108	530 Gram	530 – 534,99 Gram
109	535 Gram	535 – 539,99 Gram
110	540 Gram	540 – 544,99 Gram
111	545 Gram	545 – 549,99 Gram
112	550 Gram	550 – 554,99 Gram
113	555 Gram	555 – 559,99 Gram
114	560 Gram	560 – 564,99 Gram
115	565 Gram	565 – 569,99 Gram
116	570 Gram	570 – 574,99 Gram
117	575 Gram	575 – 579,99 Gram
118	580 Gram	580 – 584,99 Gram
119	585 Gram	585 – 589,99 Gram
120	590 Gram	590 – 594,99 Gram
121	595 Gram	595 – 599,99 Gram
122	600 Gram	600 – 604,99 Gram
123	605 Gram	605 – 609,99 Gram
124	610 Gram	610 – 614,99 Gram
125	615 Gram	615 – 619,99 Gram
126	620 Gram	620 – 624,99 Gram
127	625 Gram	625 – 629,99 Gram

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan dari alat yang sudah dibuat dan data yang diperoleh dalam pengujian maka penulis dapat menyimpulkan bahwa:

1. Desain untuk perancangan mekanik penempatan sensor *load cell* harus benar-benar sesuai dengan posisinya, sehingga mudah ditempatkan, dirakit dan mampu bekerja dengan maksimal.
2. Sensor *load cell* yang digunakan adalah *type CZL 635* yang mempunyai kelemahan lentur sel beban balok bahwa *strain gauge* biasanya terkena membutuhkan perlindungan yang cukup.
3. Timbangan *digital* berbasis Arduino Uno dengan *Output* Suara dapat bekerja secara maksimal tanpa harus melihat berapa keluar berat buah, karena timbangan ini dilengkapi dengan suara. Sehingga pembeli dapat tahu berapa berat buah tersebut. Tetapi kelemahan dari *load cell* adalah nilai yang keluar kurang efisien, sehingga butuh waktu banyak untuk mencari nilai yang mendekati atau presisi dengan nilai yang asli.
4. *Df Player* akan bekerja kalau mp3 sudah di *save* dalam *memori card*.

5.2 Saran

Penulis menyadari bahwa alat ini masih sangat membutuhkan banyak pengembangan baik dari segi penggunaan dan sistem kerja, maka penulis mempunyai beberapa saran demi kemajuan dan pengembangan alat ini yakni:

1. Pembuatan mekanik harus sesuai dengan perancangan yang dibuat, karena akan mempengaruhi sensor *load cell* apabila mekanik tidak sesuai yang diharapkan.
2. Sensor *load cell* yang digunakan haruslah benar-benar sensor yang akurat dengan *type* yang lebih stabil dan sesuai dengan bentuk mekaniknya. *Type* sensor *load cell* yang standar dengan tingkat kesetabilan yang akurat.
3. Nilai berat buah ditampilkan di LCD. Untuk pengembangan kedepan seharusnya bisa ditampilkan di layar *monitor* supaya lebih *familier*.
4. *Load cell* tidak hanya digunakan untuk mengukur berat buah saja, sebaiknya di gunakan juga pada berat badan manusia, sehingga manusia dapat tahu berapa berat badan manusia.
5. *Df player* tidak hanya digunakan untuk menyimpan satuan gram-kg. Sebaiknya digunakan juga untuk membuat permainan menggunakan *Df player*.

DAFTAR PUSTAKA

- Mapaung. M., dan Warman. E (2015) Perancangan Sistem Pengontrolan Pengukuran Berat Badan Pada Timbangan kendaraan Secara Otomatis. *Singuda Ensikom*, 10(27)53-58
- Yandra. E.F., Lapanporo. B.P., dan Jumarang. M.H (2016) Rancang Bangun Timbangan Digital Berbasis Sensor Beban 5 kg Menggunakan Mikrokontroler ATM328. *Positron* 4(1)23-28
- Aghili. F (2010) *Design of a Load Cell With Large Overload Capacity*. *Transactions of the Canadian Society for Mechanical Engineering*, 34(3-4)449-461
- Elektronika Dasar [Online] Tersedia (<http://elektronika-dasar.web.id/teori-elektronika/lcd-liquid-cristal-display/>), diakses pada tanggal 27 September 2018.
- Indoware [Online] Tersedia (<http://indo-ware.com/blog-27-timbangan-5kg-hx711.html>) diakses pada tanggal 27 September 2018
- Load Cell Teori/Load Cell Handbook [Online] Tersedia (<https://loadcellteori.wordpress.com/2011/04/05/load-cell-teori-manual-load-cell/>) diakses pada tanggal 27 September 2018.
- Kelas Mikrokontrol.(2012). *Mikrokontroler ARM*. [Online] Tersedia (<http://www.kelas-mikrokontrol.com/e-learning/mikrokontroler/mengenal-arm-cortex%C2%ADm0.html>) diakses pada tanggal 27 September 2018
- <http://www.kelas-mikrokontrol.com/e-learning/mikrokontroler/mengenal-arm-cortex%C2%ADm0.html> diakses pada tanggal 27 September 2018
- Manual DT-ARM NUC120 Board.pdf diakses pada tanggal 27 September 2018
- <https://rohmedi.com/2015/08/12/timbangan-5kg-hx711/> diakses pada tanggal 27 September 2018
- <http://elektronika-dasar.web.id/lcd-liquid-cristal-display/> diakses pada tanggal 27 September 2018

LAMPIRAN

A. Gambar alat keseluruhan



```

#ifndef __HX711__H__
#define __HX711__H__

#include <Arduino.h>

#define HX711_SCK A5
#define HX711_DT A4

#include <SoftwareSerial.h>

#define ARDUINO_RX 5 //should connect to
TX of the Serial MP3 Player module

#define ARDUINO_TX 6 //connect to RX of
the module

#include <Wire.h>
#include <LiquidCrystal.h>

// initialize the library by associating any
needed LCD interface pin

// with the arduino pin number it is connected
to

const int rs = 4, en = 7, d4 = 8, d5 = 9, d6 = 10,
d7 = 11;

LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

SoftwareSerial mp3(ARDUINO_RX,
ARDUINO_TX);

static int8_t Send_buf[8] = {0}; // Buffer for
Send commands. // BETTER LOCALLY

static uint8_t ansbuf[10] = {0}; // Buffer for
the answers. // BETTER LOCALLY

String mp3Answer; // Answer from the
MP3.

String sanswer(void);

String sbyte2hex(uint8_t b);

/***** Command byte
*****/

#define CMD_NEXT_SONG 0X01 // Play
next song.

#define CMD_PREV_SONG 0X02 // Play
previous song.

#define CMD_PLAY_W_INDEX 0X03

#define CMD_VOLUME_UP 0X04

#define CMD_VOLUME_DOWN 0X05

#define CMD_SET_VOLUME 0X06

#define CMD_SNG_CYCL_PLAY 0X08 // Single
Cycle Play.

#define CMD_SEL_DEV 0X09

#define CMD_SLEEP_MODE 0X0A

#define CMD_WAKE_UP 0X0B

#define CMD_RESET 0X0C

#define CMD_PLAY 0X0D

#define CMD_PAUSE 0X0E

```

```

#define CMD_PLAY_FOLDER_FILE 0X0F

#define CMD_STOP_PLAY 0X16 // Stop
playing continuously.

#define CMD_FOLDER_CYCLE 0X17

#define CMD_SHUFFLE_PLAY 0x18 //

#define CMD_SET_SNGL_CYCL 0X19 // Set
single cycle.

#define CMD_SET_DAC 0X1A

#define DAC_ON 0X00

#define DAC_OFF 0X01

#define CMD_PLAY_W_VOL 0X22

#define CMD_PLAYING_N 0x4C

#define CMD_QUERY_STATUS 0x42

#define CMD_QUERY_VOLUME 0x43

#define CMD_QUERY_FLDR_TRACKS 0x4e

#define CMD_QUERY_TOT_TRACKS 0x48

#define CMD_QUERY_FLDR_COUNT 0x4f

/***** Opitons
*****/

#define DEV_TF 0X02

/*****
*****/

extern void Init_Hx711();

extern unsigned long HX711_Read(void);

extern unsigned int Get_Weight();

extern void Get_Maopi();

#endif

////////////////////////////////////

//file HX711.cpp boleh dipisah

long HX711_Buffer = 0;

long Weight_Maopi = 0,Weight_Shivu = 0;

void Init_Hx711()

{

pinMode(HX711_SCK, OUTPUT);

pinMode(HX711_DT, INPUT);

}

void Get_Maopi()

{

HX711_Buffer = HX711_Read();

Weight_Maopi = HX711_Buffer/100;

}

unsigned int Get_Weight()

{

HX711_Buffer = HX711_Read();

```

```

HX711_Buffer = HX711_Buffer/100;

Weight_Shivu = HX711_Buffer;

// Weight_Shivu = Weight_Shivu -
Weight_Maopi;

// Weight_Shivu = (unsigned
int)((float)Weight_Shivu/7.35+0.05);

return Weight_Shivu;
}

```

```

unsigned long HX711_Read(void)
{
    unsigned long count;
    unsigned char i;
    bool Flag = 0;

```

```

digitalWrite(HX711_DT, HIGH);
delayMicroseconds(1);

digitalWrite(HX711_SCK, LOW);
delayMicroseconds(1);

count=0;
while(digitalRead(HX711_DT));
for(i=0;i<24;i++)
{
    digitalWrite(HX711_SCK, HIGH);
    delayMicroseconds(1);

```

```

count=count<<1;
digitalWrite(HX711_SCK, LOW);
delayMicroseconds(1);
if(digitalRead(HX711_DT))
    count++;
}
digitalWrite(HX711_SCK, HIGH);
count ^= 0x800000;
delayMicroseconds(1);
digitalWrite(HX711_SCK, LOW);
delayMicroseconds(1);

return(count);
}

////////////////////////////////////
////////////////////////////////////
int Weight = 0;
float massa = 0;
void setup() {
    sendCommand(CMD_SEL_DEV, DEV_TF);
    delay(500);
    // lcd.init();          // initialize the lcd
    // lcd.init();
    lcd.begin(16, 2);
    // Print a message to the LCD.
    // lcd.backlight();

```



```

lcd.setCursor(0,0);
lcd.print (" SELAMAT DATANG ");
delay (3000);

Serial.begin(9600);
mp3.begin(9600);
// delay(500);

// put your setup code here, to run once:
Init_Hx711();

Serial.print("selamat datang\n");

Get_Maopi();

sendCommand (CMD_SNG_CYCL_PLAY,
0x0101);

delay(1200);

sendCommand (CMD_RESET, 0x00);

lcd.clear();
}

void loop() {

// Serial.print("Tampilkan");

Weight = Get_Weight();

// massa = 0.8586*Weight+8.637;//y =
0,8586x + 8,637//massa = (0,002*Weight)-
0,005;

// if (massa >=1.00 && massa <=29.99){

massa = 0;
}

Serial.print(massa);

Serial.print(" gram\n");

Serial.print("\n");

lcd.setCursor(0,0);

lcd.print ("Berat:");

lcd.print (massa);

// lcd.home();

// lcd.setCursor(14,0);

lcd.print (" gr ");

lcd.setCursor(0,1);

lcd.print (" ");

delay(2000);

if (massa >=30.00 && massa <=34.99){ //30
gram

sendCommand (CMD_WAKE_UP, 0x00);

sendCommand (CMD_PLAY_FOLDER_FILE,
0x0101);

delay(700);

}

if (massa >=35.00 && massa <=39.99){ //35
gram

sendCommand (CMD_WAKE_UP, 0x00);
}

```

```
    sendCommand (CMD_PLAY_FOLDER_FILE,  
0x0301);
```

```
    delay(700);
```

```
}
```

```
if (massa >=40.00 && massa <=44.99){ //40  
gram
```

```
    sendCommand (CMD_WAKE_UP, 0x00);
```

```
    sendCommand (CMD_PLAY_FOLDER_FILE,  
0x0201);
```

```
    delay(700);
```

```
}
```

```
if (massa >=45.00 && massa <=49.99){ //45  
gram
```

```
    sendCommand (CMD_WAKE_UP, 0x00);
```

```
    sendCommand (CMD_PLAY_FOLDER_FILE,  
0x0401);
```

```
    delay(700);
```

```
}
```

```
if (massa >=50.00 && massa <=54.99){ //50  
gram
```

```
    sendCommand (CMD_WAKE_UP, 0x00);
```

```
    sendCommand (CMD_PLAY_FOLDER_FILE,  
0x0501);
```

```
    delay(700);
```

```
}
```

```
if (massa >=55.00 && massa <=59.99){ //55  
gram
```

```
    sendCommand (CMD_WAKE_UP, 0x00);
```

```
    sendCommand (CMD_PLAY_FOLDER_FILE,  
0x0601);
```

```
    delay(700);
```

```
}
```

```
if (massa >=60.00 && massa <=64.99){ //60  
gram
```

```
    sendCommand (CMD_WAKE_UP, 0x00);
```

```
    sendCommand (CMD_PLAY_FOLDER_FILE,  
0x0801);
```

```
    delay(700);
```

```
}
```

```
if (massa >=65.00 && massa <=69.99){ //65  
gram
```

```
    sendCommand (CMD_WAKE_UP, 0x00);
```

```
    sendCommand (CMD_PLAY_FOLDER_FILE,  
0x0701);
```

```
    delay(700);
```

```
}
```

```

if (massa >=70.00 && massa <=74.99){ //70
gram
    sendCommand (CMD_WAKE_UP, 0x00);
    sendCommand (CMD_PLAY_FOLDER_FILE,
0x0901);
    delay(700);
}

if (massa >=75.00 && massa <=79.99){ //75
gram
    sendCommand (CMD_WAKE_UP, 0x00);
    sendCommand (CMD_PLAY_FOLDER_FILE,
0x0C01);
    delay(700);
}

if (massa >=80.00 && massa <=84.99){ //80
gram
    sendCommand (CMD_WAKE_UP, 0x00);
    sendCommand (CMD_PLAY_FOLDER_FILE,
0x0D01);
    delay(700);
}

if (massa >=85.00 && massa <=89.99){ //85
gram
    sendCommand (CMD_WAKE_UP, 0x00);
    sendCommand (CMD_PLAY_FOLDER_FILE,
0x0A01);
    delay(700);
}

if (massa >=90.00 && massa <=94.99){ //90
gram
    sendCommand (CMD_WAKE_UP, 0x00);
    sendCommand (CMD_PLAY_FOLDER_FILE,
0x0B01);
    delay(700);
    //
}

if (massa >=95.00 && massa <=99.99){ //95
gram
    sendCommand (CMD_WAKE_UP, 0x00);
    sendCommand (CMD_PLAY_FOLDER_FILE,
0x0E01);
    delay(700);
}

if (massa >=100.00 && massa <=104.99){
//100 gram
    sendCommand (CMD_WAKE_UP, 0x00);
    sendCommand (CMD_PLAY_FOLDER_FILE,
0x0F01);
    delay(700);
}

```

```

}

if (massa >=105.00 && massa <=109.99){
//105 gram

sendCommand (CMD_WAKE_UP, 0x00);

sendCommand (CMD_PLAY_FOLDER_FILE,
0x1001);

delay(700);
}

if (massa >=110.00 && massa <=114.99){
//110 gram

sendCommand (CMD_WAKE_UP, 0x00);

sendCommand (CMD_PLAY_FOLDER_FILE,
0x1101);

delay(700);
}

if (massa >=115.00 && massa <=119.99){
//115 gram

sendCommand (CMD_WAKE_UP, 0x00);

sendCommand (CMD_PLAY_FOLDER_FILE,
0x1201);

delay(700);
}

if (massa >=120.00 && massa <=124.99){
//120 gram

sendCommand (CMD_WAKE_UP, 0x00);

sendCommand (CMD_PLAY_FOLDER_FILE,
0x1301);

delay(700);
}

if (massa >=125.00 && massa <=129.99){
//125 gram

sendCommand (CMD_WAKE_UP, 0x00);

sendCommand (CMD_PLAY_FOLDER_FILE,
0x1401);

delay(700);
}

if (massa >=130.00 && massa <=134.99){
//130 gram

sendCommand (CMD_WAKE_UP, 0x00);

sendCommand (CMD_PLAY_FOLDER_FILE,
0x1501);

delay(700);
}

if (massa >=135.00 && massa <=139.99){
//135 gram

sendCommand (CMD_WAKE_UP, 0x00);
}

```

```

    sendCommand (CMD_PLAY_FOLDER_FILE,
0x1601);
    delay(700);
}

if (massa >=140.00 && massa <=144.99){
//140 gram

    sendCommand (CMD_WAKE_UP, 0x00);

    sendCommand (CMD_PLAY_FOLDER_FILE,
0x1701);

    delay(700);
}

if (massa >=145.00 && massa <=149.99){
//145 gram

    sendCommand (CMD_WAKE_UP, 0x00);

    sendCommand (CMD_PLAY_FOLDER_FILE,
0x1801);

    delay(700);
}

if (massa >=150.00 && massa <=154.99){
//150 gram

    sendCommand (CMD_WAKE_UP, 0x00);

    sendCommand (CMD_PLAY_FOLDER_FILE,
0x1901);

    delay(700);
}

}

if (massa >=155.00 && massa <=159.99){
//155 gram

    sendCommand (CMD_WAKE_UP, 0x00);

    sendCommand (CMD_PLAY_FOLDER_FILE,
0x1D01);

    delay(700);
}

if (massa >=160.00 && massa <=164.99){
//160 gram

    sendCommand (CMD_WAKE_UP, 0x00);

    sendCommand (CMD_PLAY_FOLDER_FILE,
0x1E01);

    delay(700);
}

if (massa >=165.00 && massa <=169.99){
//165 gram

    sendCommand (CMD_WAKE_UP, 0x00);

    sendCommand (CMD_PLAY_FOLDER_FILE,
0x1C01);

    delay(700);
}
}

```

```
if (massa >=170.00 && massa <=174.99){
//170 gram
    sendCommand (CMD_WAKE_UP, 0x00);
    sendCommand (CMD_PLAY_FOLDER_FILE,
0x1B01);
    delay(700);
}
```

```
if (massa >=175.00 && massa <=179.99){
//175 gram
    sendCommand (CMD_WAKE_UP, 0x00);
    sendCommand (CMD_PLAY_FOLDER_FILE,
0x1F01);
    delay(700);
}
```

```
if (massa >=180.00 && massa <=184.99){
//180 gram
    sendCommand (CMD_WAKE_UP, 0x00);
    sendCommand (CMD_PLAY_FOLDER_FILE,
0x2001);
    delay(700);
}
```

```
if (massa >=185.00 && massa <=189.99){
//185 gram
////////////////////////////////////
////
```

```
    sendCommand (CMD_WAKE_UP, 0x00);
    sendCommand (CMD_PLAY_FOLDER_FILE,
0xC401);
    delay(700);
}
```

```
if (massa >=190.00 && massa <=194.99){
//190 gram
    sendCommand (CMD_WAKE_UP, 0x00);
    sendCommand (CMD_PLAY_FOLDER_FILE,
0x2101);
    delay(700);
}
```

```
if (massa >=195.00 && massa <=199.99){
//195 gram
    sendCommand (CMD_WAKE_UP, 0x00);
    sendCommand (CMD_PLAY_FOLDER_FILE,
0x2201);
    delay(700);
}
```

```
if (massa >=200.00 && massa <=204.99){
//200 gram
    sendCommand (CMD_WAKE_UP, 0x00);
    sendCommand (CMD_PLAY_FOLDER_FILE,
0x2301);
```

```

delay(700);

}

if (massa >=205.00 && massa <=209.99){
//205 gram

sendCommand (CMD_WAKE_UP, 0x00);

sendCommand (CMD_PLAY_FOLDER_FILE,
0x2401);

delay(700);

}

if (massa >=210.00 && massa <=214.99){
//210 gram

sendCommand (CMD_WAKE_UP, 0x00);

sendCommand (CMD_PLAY_FOLDER_FILE,
0x2501);

delay(700);

}

if (massa >=215.00 && massa <=219.99){
//215 gram

sendCommand (CMD_WAKE_UP, 0x00);

sendCommand (CMD_PLAY_FOLDER_FILE,
0x2601);

delay(700);

}

if (massa >=220.00 && massa <=224.99){
//220 gram

sendCommand (CMD_WAKE_UP, 0x00);

sendCommand (CMD_PLAY_FOLDER_FILE,
0x2B01);

delay(700);

}

if (massa >=225.00 && massa <=229.99){
//225 gram

sendCommand (CMD_WAKE_UP, 0x00);

sendCommand (CMD_PLAY_FOLDER_FILE,
0x2701);

delay(700);

}

if (massa >=230.00 && massa <=234.99){
//230 gram

sendCommand (CMD_WAKE_UP, 0x00);

sendCommand (CMD_PLAY_FOLDER_FILE,
0x2801);

delay(700);

}

if (massa >=235.00 && massa <=239.99){
//235 gram

```

```

    sendCommand (CMD_WAKE_UP, 0x00);          delay(700);

    sendCommand (CMD_PLAY_FOLDER_FILE,
0x2A01);
}

    delay(700);

}

    if (massa >=255.00 && massa <=259.99){
//255 gram

        sendCommand (CMD_WAKE_UP, 0x00);

        sendCommand (CMD_PLAY_FOLDER_FILE,
0x2E01);

        delay(700);

    }

    if (massa >=240.00 && massa <=244.99){
//240 gram

        sendCommand (CMD_WAKE_UP, 0x00);          delay(700);

        sendCommand (CMD_PLAY_FOLDER_FILE,
0x2901);

        delay(700);

    }

    if (massa >=260.00 && massa <=264.99){
//260 gram

        sendCommand (CMD_WAKE_UP, 0x00);

        sendCommand (CMD_PLAY_FOLDER_FILE,
0x2F01);

        delay(700);

    }

    if (massa >=245.00 && massa <=249.99){
//245 gram

        sendCommand (CMD_WAKE_UP, 0x00);          delay(700);

        sendCommand (CMD_PLAY_FOLDER_FILE,
0x2C01);

        delay(700);

    }

    if (massa >=265.00 && massa <=269.99){
//265 gram

        sendCommand (CMD_WAKE_UP, 0x00);

        sendCommand (CMD_PLAY_FOLDER_FILE,
0x3001);

        delay(700);

    }

    if (massa >=250.00 && massa <=254.99){
//250 gram

        sendCommand (CMD_WAKE_UP, 0x00);          delay(700);

        sendCommand (CMD_PLAY_FOLDER_FILE,
0x2D01);

    }

```



```

        sendCommand (CMD_WAKE_UP, 0x00);

        if (massa >=270.00 && massa <=274.99){
//270 gram

        sendCommand (CMD_WAKE_UP, 0x00);

        sendCommand (CMD_PLAY_FOLDER_FILE,
0x3101);

        delay(700);

        }

        if (massa >=275.00 && massa <=279.99){
//275 gram

        sendCommand (CMD_WAKE_UP, 0x00);

        sendCommand (CMD_PLAY_FOLDER_FILE,
0x3201);

        delay(700);

        }

        if (massa >=280.00 && massa <=284.99){
//280 gram

        sendCommand (CMD_WAKE_UP, 0x00);

        sendCommand (CMD_PLAY_FOLDER_FILE,
0x3401);

        delay(700);

        }

        if (massa >=285.00 && massa <=289.99){
//285 gram

        sendCommand (CMD_WAKE_UP, 0x00);

        sendCommand (CMD_PLAY_FOLDER_FILE,
0x3301);

        delay(700);

        }

        if (massa >=290.00 && massa <=294.99){
//290 gram

        sendCommand (CMD_WAKE_UP, 0x00);

        sendCommand (CMD_PLAY_FOLDER_FILE,
0x3501);

        delay(700);

        }

        if (massa >=295.00 && massa <=299.99){
//295 gram

        sendCommand (CMD_WAKE_UP, 0x00);

        sendCommand (CMD_PLAY_FOLDER_FILE,
0x3601);

        delay(700);

        }

        if (massa >=300.00 && massa <=304.99){
//300 gram

        sendCommand (CMD_WAKE_UP, 0x00);

        sendCommand (CMD_PLAY_FOLDER_FILE,
0x3701);

```

```

delay(700);

}

if (massa >=305.00 && massa <=309.99){
//305 gram

sendCommand (CMD_WAKE_UP, 0x00);

sendCommand (CMD_PLAY_FOLDER_FILE,
0x3801);

delay(700);

}

if (massa >=310.00 && massa <=314.99){
//310 gram

sendCommand (CMD_WAKE_UP, 0x00);

sendCommand (CMD_PLAY_FOLDER_FILE,
0x3901);

delay(700);

}

if (massa >=315.00 && massa <=319.99){
//315 gram

sendCommand (CMD_WAKE_UP, 0x00);

sendCommand (CMD_PLAY_FOLDER_FILE,
0x3A01);

delay(700);

}

if (massa >=320.00 && massa <=324.99){
//320 gram

sendCommand (CMD_WAKE_UP, 0x00);

sendCommand (CMD_PLAY_FOLDER_FILE,
0x3B01);

delay(700);

}

if (massa >=325.00 && massa <=329.99){
//325 gram

sendCommand (CMD_WAKE_UP, 0x00);

sendCommand (CMD_PLAY_FOLDER_FILE,
0x4001);

delay(700);

}

if (massa >=330.00 && massa <=334.99){
//330 gram

sendCommand (CMD_WAKE_UP, 0x00);

sendCommand (CMD_PLAY_FOLDER_FILE,
0x4501);

delay(700);

}

if (massa >=335.00 && massa <=339.99){
//335 gram

```

```

    sendCommand (CMD_WAKE_UP, 0x00);          delay(700);

    sendCommand (CMD_PLAY_FOLDER_FILE,
0x4601);
}

    delay(700);

}

    if (massa >=355.00 && massa <=359.99){
//355 gram

        sendCommand (CMD_WAKE_UP, 0x00);

        sendCommand (CMD_PLAY_FOLDER_FILE,
0x4101);

        delay(700);

    }

    if (massa >=340.00 && massa <=344.99){
//340 gram

        sendCommand (CMD_WAKE_UP, 0x00);          delay(700);

        sendCommand (CMD_PLAY_FOLDER_FILE,
0x4701);

        delay(700);

    }

    if (massa >=360.00 && massa <=364.99){
//360 gram

        sendCommand (CMD_WAKE_UP, 0x00);

        sendCommand (CMD_PLAY_FOLDER_FILE,
0x3E01);

        delay(700);

    }

    if (massa >=345.00 && massa <=349.99){
//345 gram

        sendCommand (CMD_WAKE_UP, 0x00);          delay(700);

        sendCommand (CMD_PLAY_FOLDER_FILE,
0x3C01);

        delay(700);

    }

    if (massa >=365.00 && massa <=369.99){
//365 gram

        sendCommand (CMD_WAKE_UP, 0x00);

        sendCommand (CMD_PLAY_FOLDER_FILE,
0x4201);

        delay(700);

    }

    if (massa >=350.00 && massa <=354.99){
//350 gram

        sendCommand (CMD_WAKE_UP, 0x00);          delay(700);

        sendCommand (CMD_PLAY_FOLDER_FILE,
0x3D01);

    }

```

```

        sendCommand (CMD_WAKE_UP, 0x00);

        if (massa >=370.00 && massa <=374.99){
//370 gram

        sendCommand (CMD_WAKE_UP, 0x00);

        sendCommand (CMD_PLAY_FOLDER_FILE,
0x4301);

        delay(700);

        }

        if (massa >=375.00 && massa <=379.99){
//375 gram

        sendCommand (CMD_WAKE_UP, 0x00);

        sendCommand (CMD_PLAY_FOLDER_FILE,
0x4401);

        delay(700);

        }

        if (massa >=380.00 && massa <=384.99){
//380 gram

        sendCommand (CMD_WAKE_UP, 0x00);

        sendCommand (CMD_PLAY_FOLDER_FILE,
0x4801);

        delay(700);

        }

        if (massa >=385.00 && massa <=389.99){
//385 gram

        sendCommand (CMD_WAKE_UP, 0x00);

        sendCommand (CMD_PLAY_FOLDER_FILE,
0x4901);

        delay(700);

        }

        if (massa >=390.00 && massa <=394.99){
//390 gram

        sendCommand (CMD_WAKE_UP, 0x00);

        sendCommand (CMD_PLAY_FOLDER_FILE,
0x4A01);

        delay(700);

        }

        if (massa >=395.00 && massa <=399.99){
//395 gram

        sendCommand (CMD_WAKE_UP, 0x00);

        sendCommand (CMD_PLAY_FOLDER_FILE,
0x4B01);

        delay(700);

        }

        if (massa >=400.00 && massa <=404.99){
//400 gram

        sendCommand (CMD_WAKE_UP, 0x00);

        sendCommand (CMD_PLAY_FOLDER_FILE,
0x3F01);

```

```

delay(700);

}

if (massa >=405.00 && massa <=409.99){
//405 gram

sendCommand (CMD_WAKE_UP, 0x00);

sendCommand (CMD_PLAY_FOLDER_FILE,
0x4C01);

delay(700);

}

if (massa >=410.00 && massa <=414.99){
//410 gram

sendCommand (CMD_WAKE_UP, 0x00);

sendCommand (CMD_PLAY_FOLDER_FILE,
0x5201);

delay(700);

}

if (massa >=415.00 && massa <=419.99){
//415 gram

sendCommand (CMD_WAKE_UP, 0x00);

sendCommand (CMD_PLAY_FOLDER_FILE,
0x5101);

delay(700);

}

if (massa >=420.00 && massa <=424.99){
//420 gram

sendCommand (CMD_WAKE_UP, 0x00);

sendCommand (CMD_PLAY_FOLDER_FILE,
0x4D01);

delay(700);

}

if (massa >=425.00 && massa <=429.99){
//425 gram

sendCommand (CMD_WAKE_UP, 0x00);

sendCommand (CMD_PLAY_FOLDER_FILE,
0x5301);

delay(700);

}

if (massa >=430.00 && massa <=434.99){
//430 gram

sendCommand (CMD_WAKE_UP, 0x00);

sendCommand (CMD_PLAY_FOLDER_FILE,
0x4E01);

delay(700);

}

if (massa >=435.00 && massa <=439.99){
//435 gram

```

```

    sendCommand (CMD_WAKE_UP, 0x00);          delay(700);

    sendCommand (CMD_PLAY_FOLDER_FILE,
0x4F01);
}

    delay(700);

}

    if (massa >=455.00 && massa <=459.99){
//455 gram

        sendCommand (CMD_WAKE_UP, 0x00);

        sendCommand (CMD_PLAY_FOLDER_FILE,
0x5701);

        delay(700);

    }

    if (massa >=440.00 && massa <=444.99){
//440 gram

        sendCommand (CMD_WAKE_UP, 0x00);

        sendCommand (CMD_PLAY_FOLDER_FILE,
0x5401);

        delay(700);

    }

    if (massa >=460.00 && massa <=464.99){
//460 gram

        sendCommand (CMD_WAKE_UP, 0x00);

        sendCommand (CMD_PLAY_FOLDER_FILE,
0x5001);

        delay(700);

    }

    if (massa >=445.00 && massa <=449.99){
//445 gram

        sendCommand (CMD_WAKE_UP, 0x00);

        sendCommand (CMD_PLAY_FOLDER_FILE,
0x5501);

        delay(700);

    }

    if (massa >=465.00 && massa <=469.99){
//465 gram

        sendCommand (CMD_WAKE_UP, 0x00);

        sendCommand (CMD_PLAY_FOLDER_FILE,
0xAC01);

        delay(700);

    }

    if (massa >=450.00 && massa <=454.99){
//450 gram

        sendCommand (CMD_WAKE_UP, 0x00);

        sendCommand (CMD_PLAY_FOLDER_FILE,
0x5601);

    }

```

```

        sendCommand (CMD_WAKE_UP, 0x00);

        if (massa >=470.00 && massa <=474.99){
//470 gram

        sendCommand (CMD_WAKE_UP, 0x00);

        sendCommand (CMD_PLAY_FOLDER_FILE,
0x5801);

        delay(700);

        }

        if (massa >=475.00 && massa <=479.99){
//475 gram

        sendCommand (CMD_WAKE_UP, 0x00);

        sendCommand (CMD_PLAY_FOLDER_FILE,
0x5901);

        delay(700);

        }

        if (massa >=480.00 && massa <=484.99){
//480 gram

        sendCommand (CMD_WAKE_UP, 0x00);

        sendCommand (CMD_PLAY_FOLDER_FILE,
0x5A01);

        delay(700);

        }

        if (massa >=485.00 && massa <=489.99){
//485 gram

        sendCommand (CMD_WAKE_UP, 0x00);

        sendCommand (CMD_PLAY_FOLDER_FILE,
0x5B01);

        delay(700);

        }

        if (massa >=490.00 && massa <=494.99){
//490 gram

        sendCommand (CMD_WAKE_UP, 0x00);

        sendCommand (CMD_PLAY_FOLDER_FILE,
0x5C01);

        delay(700);

        }

        if (massa >=495.00 && massa <=499.99){
//495 gram

        sendCommand (CMD_WAKE_UP, 0x00);

        sendCommand (CMD_PLAY_FOLDER_FILE,
0x5D01);

        delay(700);

        }

        if (massa >=500.00 && massa <=504.99){
//500 gram

        sendCommand (CMD_WAKE_UP, 0x00);

        sendCommand (CMD_PLAY_FOLDER_FILE,
0x5E01);

```

```

delay(700);

}

if (massa >=505.00 && massa <=509.99){
//505 gram

sendCommand (CMD_WAKE_UP, 0x00);

sendCommand (CMD_PLAY_FOLDER_FILE,
0x6101);

delay(700);

}

if (massa >=510.00 && massa <=514.99){
//510 gram

sendCommand (CMD_WAKE_UP, 0x00);

sendCommand (CMD_PLAY_FOLDER_FILE,
0x5F01);

delay(700);

}

if (massa >=515.00 && massa <=519.99){
//515 gram

sendCommand (CMD_WAKE_UP, 0x00);

sendCommand (CMD_PLAY_FOLDER_FILE,
0x6001);

delay(700);

}

if (massa >=520.00 && massa <=524.99){
//520 gram

sendCommand (CMD_WAKE_UP, 0x00);

sendCommand (CMD_PLAY_FOLDER_FILE,
0x6201);

delay(700);

}

if (massa >=525.00 && massa <=529.99){
//525 gram

sendCommand (CMD_WAKE_UP, 0x00);

sendCommand (CMD_PLAY_FOLDER_FILE,
0x6301);

delay(700);

}

/*****
***** BATAS FOLDER 99
*****
*****/

if (massa >=530.00 && massa <=534.99){
//530 gram

sendCommand (CMD_WAKE_UP, 0x00);

sendCommand (CMD_PLAY_FOLDER_FILE,
0x0102); //Folder 100

delay(700);

}

```



```

}

if (massa >=535.00 && massa <=539.99){
//535 gram
////////////////////////////////////

sendCommand (CMD_WAKE_UP, 0x00);

sendCommand (CMD_PLAY_FOLDER_FILE,
0x0202);

delay(700);

}

if (massa >=540.00 && massa <=544.99){
//540 gram

sendCommand (CMD_WAKE_UP, 0x00);

sendCommand (CMD_PLAY_FOLDER_FILE,
0x0302);

delay(700);

}

if (massa >=545.00 && massa <=549.99){
//545 gram

sendCommand (CMD_WAKE_UP, 0x00);

sendCommand (CMD_PLAY_FOLDER_FILE,
0x0402);

delay(700);

}

if (massa >=550.00 && massa <=554.99){
//550 gram

sendCommand (CMD_WAKE_UP, 0x00);

sendCommand (CMD_PLAY_FOLDER_FILE,
0x0502);

delay(700);

}

if (massa >=555.00 && massa <=559.99){
//555 gram

sendCommand (CMD_WAKE_UP, 0x00);

sendCommand (CMD_PLAY_FOLDER_FILE,
0x0602);

delay(700);

}

if (massa >=560.00 && massa <=564.99){
//560 gram

sendCommand (CMD_WAKE_UP, 0x00);

sendCommand (CMD_PLAY_FOLDER_FILE,
0x0702);

delay(700);

}

if (massa >=565.00 && massa <=569.99){
//565 gram

sendCommand (CMD_WAKE_UP, 0x00);

```

```

    sendCommand (CMD_PLAY_FOLDER_FILE,
0x0802);
    delay(700);
}

if (massa >=570.00 && massa <=574.99){
//570 gram

    sendCommand (CMD_WAKE_UP, 0x00);

    sendCommand (CMD_PLAY_FOLDER_FILE,
0x0902);

    delay(700);
}

if (massa >=575.00 && massa <=579.99){
//575 gram

    sendCommand (CMD_WAKE_UP, 0x00);

    sendCommand (CMD_PLAY_FOLDER_FILE,
0x0A02);

    delay(700);
}

if (massa >=580.00 && massa <=584.99){
//580 gram

    sendCommand (CMD_WAKE_UP, 0x00);

    sendCommand (CMD_PLAY_FOLDER_FILE,
0x0B02);

    delay(700);
}

}

if (massa >=585.00 && massa <=589.99){
//585 gram

    sendCommand (CMD_WAKE_UP, 0x00);

    sendCommand (CMD_PLAY_FOLDER_FILE,
0x0C02);

    delay(700);
}

if (massa >=590.00 && massa <=594.99){
//590 gram

    sendCommand (CMD_WAKE_UP, 0x00);

    sendCommand (CMD_PLAY_FOLDER_FILE,
0x0D02);

    delay(700);
}

if (massa >=595.00 && massa <=599.99){
//595 gram

    sendCommand (CMD_WAKE_UP, 0x00);

    sendCommand (CMD_PLAY_FOLDER_FILE,
0x0E02);

    delay(700);
}
}

```

```

if (massa >=600.00 && massa <=604.99){
//600 gram
    sendCommand (CMD_WAKE_UP, 0x00);
    sendCommand (CMD_PLAY_FOLDER_FILE,
0x0F02);
    delay(700);
}

if (massa >=605.00 && massa <=609.99){
//605 gram
    sendCommand (CMD_WAKE_UP, 0x00);
    sendCommand (CMD_PLAY_FOLDER_FILE,
0x1002);
    delay(700);
}

if (massa >=610.00 && massa <=614.99){
//610 gram
    sendCommand (CMD_WAKE_UP, 0x00);
    sendCommand (CMD_PLAY_FOLDER_FILE,
0x1102);
    delay(700);
}

if (massa >=615.00 && massa <=619.99){
//615 gram
    sendCommand (CMD_WAKE_UP, 0x00);
    sendCommand (CMD_PLAY_FOLDER_FILE,
0x1202);
    delay(700);
}

if (massa >=620.00 && massa <=624.99){
//620 gram
    sendCommand (CMD_WAKE_UP, 0x00);
    sendCommand (CMD_PLAY_FOLDER_FILE,
0x1302);
    delay(700);
}

if (massa >=625.00 && massa <=629.99){
//625 gram
    sendCommand (CMD_WAKE_UP, 0x00);
    sendCommand (CMD_PLAY_FOLDER_FILE,
0x1402);
    delay(700);
}

if (massa >=630.00 && massa <=634.99){
//630 gram
    sendCommand (CMD_WAKE_UP, 0x00);
    sendCommand (CMD_PLAY_FOLDER_FILE,
0x1502);
    delay(700);
}

```

```

}

if (massa >=635.00 && massa <=639.99){
//635 gram

    sendCommand (CMD_WAKE_UP, 0x00);

    sendCommand (CMD_PLAY_FOLDER_FILE,
0x1602);

    delay(700);
}

if (massa >=640.00 && massa <=644.99){
//640 gram

    sendCommand (CMD_WAKE_UP, 0x00);

    sendCommand (CMD_PLAY_FOLDER_FILE,
0x1702);

    delay(700);
}

if (massa >=645.00 && massa <=649.99){
//645 gram

    sendCommand (CMD_WAKE_UP, 0x00);

    sendCommand (CMD_PLAY_FOLDER_FILE,
0x1802);

    delay(700);
}

if (massa >=650.00 && massa <=654.99){
//650 gram

    sendCommand (CMD_WAKE_UP, 0x00);

    sendCommand (CMD_PLAY_FOLDER_FILE,
0x1902);

    delay(700);
}

if (massa >=655.00 && massa <=659.99){
//655 gram

    sendCommand (CMD_WAKE_UP, 0x00);

    sendCommand (CMD_PLAY_FOLDER_FILE,
0x1A02);

    delay(700);
}

if (massa >=660.00 && massa <=664.99){
//660 gram

    sendCommand (CMD_WAKE_UP, 0x00);

    sendCommand (CMD_PLAY_FOLDER_FILE,
0x1B02);

    delay(700);
}

if (massa >=665.00 && massa <=669.99){
//665 gram

    sendCommand (CMD_WAKE_UP, 0x00);

```

```
    sendCommand (CMD_PLAY_FOLDER_FILE,  
0x1C02);
```

```
    delay(700);
```

```
}
```

```
if (massa >=670.00 && massa <=674.99){  
//670 gram
```

```
    sendCommand (CMD_WAKE_UP, 0x00);
```

```
    sendCommand (CMD_PLAY_FOLDER_FILE,  
0x1D02);
```

```
    delay(700);
```

```
}
```

```
if (massa >=675.00 && massa <=679.99){  
//675 gram
```

```
    sendCommand (CMD_WAKE_UP, 0x00);
```

```
    sendCommand (CMD_PLAY_FOLDER_FILE,  
0x1E02);
```

```
    delay(700);
```

```
}
```

```
if (massa >=680.00 && massa <=684.99){  
//680 gram
```

```
    sendCommand (CMD_WAKE_UP, 0x00);
```

```
    sendCommand (CMD_PLAY_FOLDER_FILE,  
0x1F02);
```

```
    delay(700);
```

```
}
```

```
if (massa >=685.00 && massa <=689.99){  
//685 gram
```

```
    sendCommand (CMD_WAKE_UP, 0x00);
```

```
    sendCommand (CMD_PLAY_FOLDER_FILE,  
0x2002);
```

```
    delay(700);
```

```
}
```

```
if (massa >=690.00 && massa <=694.99){  
//690 gram
```

```
    sendCommand (CMD_WAKE_UP, 0x00);
```

```
    sendCommand (CMD_PLAY_FOLDER_FILE,  
0x2102);
```

```
    delay(700);
```

```
}
```

```
if (massa >=695.00 && massa <=699.99){  
//695 gram
```

```
    sendCommand (CMD_WAKE_UP, 0x00);
```

```
    sendCommand (CMD_PLAY_FOLDER_FILE,  
0x2202);
```

```
    delay(700);
```

```
}
```

```

if (massa >=700.00 && massa <=704.99){
//700 gram
    sendCommand (CMD_WAKE_UP, 0x00);
    sendCommand (CMD_PLAY_FOLDER_FILE,
0x2302);
    delay(700);
}

if (massa >=705.00 && massa <=709.99){
//705 gram
    sendCommand (CMD_WAKE_UP, 0x00);
    sendCommand (CMD_PLAY_FOLDER_FILE,
0x2402);
    delay(700);
}

if (massa >=710.00 && massa <=714.99){
//710 gram
    sendCommand (CMD_WAKE_UP, 0x00);
    sendCommand (CMD_PLAY_FOLDER_FILE,
0x2502);
    delay(700);
}

if (massa >=715.00 && massa <=719.99){
//715 gram
    sendCommand (CMD_WAKE_UP, 0x00);
    sendCommand (CMD_PLAY_FOLDER_FILE,
0x2602);
    delay(700);
}

if (massa >=720.00 && massa <=724.99){
//720 gram
    sendCommand (CMD_WAKE_UP, 0x00);
    sendCommand (CMD_PLAY_FOLDER_FILE,
0x2702);
    delay(700);
}

if (massa >=725.00 && massa <=729.99){
//725 gram
    sendCommand (CMD_WAKE_UP, 0x00);
    sendCommand (CMD_PLAY_FOLDER_FILE,
0x2802);
    delay(700);
}

if (massa >=730.00 && massa <=734.99){
//730 gram
    sendCommand (CMD_WAKE_UP, 0x00);
    sendCommand (CMD_PLAY_FOLDER_FILE,
0x2902);
    delay(700);
}

```

```

}

if (massa >=735.00 && massa <=739.99){
//735 gram

sendCommand (CMD_WAKE_UP, 0x00);

sendCommand (CMD_PLAY_FOLDER_FILE,
0x2A02);

delay(700);
}

if (massa >=740.00 && massa <=744.99){
//740 gram

sendCommand (CMD_WAKE_UP, 0x00);

sendCommand (CMD_PLAY_FOLDER_FILE,
0x2B02);

delay(700);
}

if (massa >=745.00 && massa <=749.99){
//745 gram

sendCommand (CMD_WAKE_UP, 0x00);

sendCommand (CMD_PLAY_FOLDER_FILE,
0x2C02);

delay(700);
}

if (massa >=750.00 && massa <=754.99){
//750 gram

sendCommand (CMD_WAKE_UP, 0x00);

sendCommand (CMD_PLAY_FOLDER_FILE,
0x2D02);

delay(700);
}

if (massa >=755.00 && massa <=759.99){
//755 gram

sendCommand (CMD_WAKE_UP, 0x00);

sendCommand (CMD_PLAY_FOLDER_FILE,
0x2E02);

delay(700);
}

if (massa >=760.00 && massa <=764.99){
//760 gram

sendCommand (CMD_WAKE_UP, 0x00);

sendCommand (CMD_PLAY_FOLDER_FILE,
0x2F02);

delay(700);
}

if (massa >=765.00 && massa <=769.99){
//765 gram

sendCommand (CMD_WAKE_UP, 0x00);

```

```
    sendCommand (CMD_PLAY_FOLDER_FILE,  
0x3002);
```

```
    delay(700);
```

```
}
```

```
if (massa >=770.00 && massa <=774.99){  
//770 gram
```

```
    sendCommand (CMD_WAKE_UP, 0x00);
```

```
    sendCommand (CMD_PLAY_FOLDER_FILE,  
0x3102);
```

```
    delay(700);
```

```
}
```

```
if (massa >=775.00 && massa <=779.99){  
//775 gram
```

```
    sendCommand (CMD_WAKE_UP, 0x00);
```

```
    sendCommand (CMD_PLAY_FOLDER_FILE,  
0x3202);
```

```
    delay(700);
```

```
}
```

```
if (massa >=780.00 && massa <=784.99){  
//780 gram
```

```
    sendCommand (CMD_WAKE_UP, 0x00);
```

```
    sendCommand (CMD_PLAY_FOLDER_FILE,  
0x3302);
```

```
    delay(700);
```

```
}
```

```
if (massa >=785.00 && massa <=789.99){  
//785 gram
```

```
    sendCommand (CMD_WAKE_UP, 0x00);
```

```
    sendCommand (CMD_PLAY_FOLDER_FILE,  
0x3402);
```

```
    delay(700);
```

```
}
```

```
if (massa >=790.00 && massa <=794.99){  
//790 gram
```

```
    sendCommand (CMD_WAKE_UP, 0x00);
```

```
    sendCommand (CMD_PLAY_FOLDER_FILE,  
0x3502);
```

```
    delay(700);
```

```
}
```

```
if (massa >=795.00 && massa <=799.99){  
//795 gram
```

```
    sendCommand (CMD_WAKE_UP, 0x00);
```

```
    sendCommand (CMD_PLAY_FOLDER_FILE,  
0x3602);
```

```
    delay(700);
```

```
}
```



```

if (massa >=800.00 && massa <=804.99){
//800 gram
    sendCommand (CMD_WAKE_UP, 0x00);
    sendCommand (CMD_PLAY_FOLDER_FILE,
0x3702);
    delay(700);
}

if (massa >=805.00 && massa <=809.99){
//805 gram
    sendCommand (CMD_WAKE_UP, 0x00);
    sendCommand (CMD_PLAY_FOLDER_FILE,
0x3802);
    delay(700);
}

if (massa >=810.00 && massa <=814.99){
//810 gram
    sendCommand (CMD_WAKE_UP, 0x00);
    sendCommand (CMD_PLAY_FOLDER_FILE,
0x3902);
    delay(700);
}

if (massa >=815.00 && massa <=819.99){
//815 gram
    sendCommand (CMD_WAKE_UP, 0x00);
    sendCommand (CMD_PLAY_FOLDER_FILE,
0x3A02);
    delay(700);
}

if (massa >=820.00 && massa <=824.99){
//820 gram
    sendCommand (CMD_WAKE_UP, 0x00);
    sendCommand (CMD_PLAY_FOLDER_FILE,
0x3B02);
    delay(700);
}

if (massa >=825.00 && massa <=829.99){
//825 gram
    sendCommand (CMD_WAKE_UP, 0x00);
    sendCommand (CMD_PLAY_FOLDER_FILE,
0x3C02);
    delay(700);
}

if (massa >=830.00 && massa <=834.99){
//830 gram
    sendCommand (CMD_WAKE_UP, 0x00);
    sendCommand (CMD_PLAY_FOLDER_FILE,
0x3D02);
    delay(700);
}

```

```

}

if (massa >=835.00 && massa <=839.99){
//835 gram

sendCommand (CMD_WAKE_UP, 0x00);

sendCommand (CMD_PLAY_FOLDER_FILE,
0x3E02);

delay(700);

}

if (massa >=840.00 && massa <=844.99){
//840 gram

sendCommand (CMD_WAKE_UP, 0x00);

sendCommand (CMD_PLAY_FOLDER_FILE,
0x3F02);

delay(700);

}

if (massa >=845.00 && massa <=849.99){
//845 gram

sendCommand (CMD_WAKE_UP, 0x00);

sendCommand (CMD_PLAY_FOLDER_FILE,
0x4002);

delay(700);

}

if (massa >=850.00 && massa <=854.99){
//850 gram

sendCommand (CMD_WAKE_UP, 0x00);

sendCommand (CMD_PLAY_FOLDER_FILE,
0x4102);

delay(700);

}

if (massa >=855.00 && massa <=859.99){
//855 gram

sendCommand (CMD_WAKE_UP, 0x00);

sendCommand (CMD_PLAY_FOLDER_FILE,
0x4202);

delay(700);

}

if (massa >=860.00 && massa <=864.99){
//860 gram

sendCommand (CMD_WAKE_UP, 0x00);

sendCommand (CMD_PLAY_FOLDER_FILE,
0x4302);

delay(700);

}

if (massa >=865.00 && massa <=869.99){
//865 gram

sendCommand (CMD_WAKE_UP, 0x00);

```

```

    sendCommand (CMD_PLAY_FOLDER_FILE,
0x4402);
    delay(700);
}

if (massa >=870.00 && massa <=874.99){
//870 gram

    sendCommand (CMD_WAKE_UP, 0x00);

    sendCommand (CMD_PLAY_FOLDER_FILE,
0x4502);

    delay(700);
}

if (massa >=875.00 && massa <=879.99){
//875 gram

    sendCommand (CMD_WAKE_UP, 0x00);

    sendCommand (CMD_PLAY_FOLDER_FILE,
0x4602);

    delay(700);
}

if (massa >=880.00 && massa <=884.99){
//880 gram

    sendCommand (CMD_WAKE_UP, 0x00);

    sendCommand (CMD_PLAY_FOLDER_FILE,
0x4702);

    delay(700);
}

}

if (massa >=885.00 && massa <=889.99){
//885 gram

    sendCommand (CMD_WAKE_UP, 0x00);

    sendCommand (CMD_PLAY_FOLDER_FILE,
0x4802);

    delay(700);
}

if (massa >=890.00 && massa <=894.99){
//890 gram

    sendCommand (CMD_WAKE_UP, 0x00);

    sendCommand (CMD_PLAY_FOLDER_FILE,
0x4902);

    delay(700);
}

if (massa >=895.00 && massa <=899.99){
//895 gram

    sendCommand (CMD_WAKE_UP, 0x00);

    sendCommand (CMD_PLAY_FOLDER_FILE,
0x4A02);

    delay(700);
}
}

```

```

if (massa >=900.00 && massa <=904.99){
//900 gram
    sendCommand (CMD_WAKE_UP, 0x00);
    sendCommand (CMD_PLAY_FOLDER_FILE,
0x4B02);
    delay(700);
}

if (massa >=905.00 && massa <=909.99){
//905 gram
    sendCommand (CMD_WAKE_UP, 0x00);
    sendCommand (CMD_PLAY_FOLDER_FILE,
0x4C02);
    delay(700);
}

if (massa >=910.00 && massa <=914.99){
//910 gram
    sendCommand (CMD_WAKE_UP, 0x00);
    sendCommand (CMD_PLAY_FOLDER_FILE,
0x4D02);
    delay(700);
}

if (massa >=915.00 && massa <=919.99){
//915 gram
    sendCommand (CMD_WAKE_UP, 0x00);
    sendCommand (CMD_PLAY_FOLDER_FILE,
0x4E02);
    delay(700);
}

if (massa >=920.00 && massa <=924.99){
//920 gram
    sendCommand (CMD_WAKE_UP, 0x00);
    sendCommand (CMD_PLAY_FOLDER_FILE,
0x4F02);
    delay(700);
}

if (massa >=925.00 && massa <=929.99){
//925 gram
    sendCommand (CMD_WAKE_UP, 0x00);
    sendCommand (CMD_PLAY_FOLDER_FILE,
0x5002);
    delay(700);
}

if (massa >=930.00 && massa <=934.99){
//930 gram
    sendCommand (CMD_WAKE_UP, 0x00);
    sendCommand (CMD_PLAY_FOLDER_FILE,
0x5102);
    delay(700);
}

```

```

}

if (massa >=935.00 && massa <=939.99){
//935 gram

sendCommand (CMD_WAKE_UP, 0x00);

sendCommand (CMD_PLAY_FOLDER_FILE,
0x5202);

delay(700);
}

if (massa >=940.00 && massa <=944.99){
//940 gram

sendCommand (CMD_WAKE_UP, 0x00);

sendCommand (CMD_PLAY_FOLDER_FILE,
0x5302);

delay(700);
}

if (massa >=945.00 && massa <=949.99){
//945 gram

sendCommand (CMD_WAKE_UP, 0x00);

sendCommand (CMD_PLAY_FOLDER_FILE,
0x5402);

delay(700);
}

if (massa >=950.00 && massa <=957.99){
//950 gram

sendCommand (CMD_WAKE_UP, 0x00);

sendCommand (CMD_PLAY_FOLDER_FILE,
0x5502);

delay(700);
}

if (massa >=955.00 && massa <=959.99){
//955 gram

sendCommand (CMD_WAKE_UP, 0x00);

sendCommand (CMD_PLAY_FOLDER_FILE,
0x5602);

delay(700);
}

if (massa >=960.00 && massa <=964.99){
//960 gram

sendCommand (CMD_WAKE_UP, 0x00);

sendCommand (CMD_PLAY_FOLDER_FILE,
0x5702);

delay(700);
}

if (massa >=965.00 && massa <=969.99){
//965 gram

sendCommand (CMD_WAKE_UP, 0x00);

```

```

    sendCommand (CMD_PLAY_FOLDER_FILE,
0x5802);
    delay(700);
}

if (massa >=970.00 && massa <=974.99){
//970 gram

    sendCommand (CMD_WAKE_UP, 0x00);

    sendCommand (CMD_PLAY_FOLDER_FILE,
0x5902);

    delay(700);
}

if (massa >=975.00 && massa <=979.99){
//975 gram

    sendCommand (CMD_WAKE_UP, 0x00);

    sendCommand (CMD_PLAY_FOLDER_FILE,
0x5A02);

    delay(700);
}

if (massa >=980.00 && massa <=984.99){
//980 gram

    sendCommand (CMD_WAKE_UP, 0x00);

    sendCommand (CMD_PLAY_FOLDER_FILE,
0x5B02);

    delay(700);
}

}

if (massa >=985.00 && massa <=989.99){
//985 gram

    sendCommand (CMD_WAKE_UP, 0x00);

    sendCommand (CMD_PLAY_FOLDER_FILE,
0x5C02);

    delay(700);
}

if (massa >=990.00 && massa <=994.99){
//990 gram

    sendCommand (CMD_WAKE_UP, 0x00);

    sendCommand (CMD_PLAY_FOLDER_FILE,
0x5D02);

    delay(700);
}

if (massa >=995.00 && massa <=999.99){
//995 gram

    sendCommand (CMD_WAKE_UP, 0x00);

    sendCommand (CMD_PLAY_FOLDER_FILE,
0x5E02);

    delay(700);
}
}

```

```

if (massa >=1000.00 && massa <=1005.00){
//1000 gram

  sendCommand (CMD_WAKE_UP, 0x00);

  sendCommand (CMD_PLAY_FOLDER_FILE,
0x5F02);

  delay(700);

}

```

```

if (massa >5000.00){ //945 gram

  lcd.setCursor(0,0);

  lcd.print ("  MAAF  ");

  lcd.setCursor(0,1);

  lcd.print (" BEBAN BERLEBIH ");

  sendCommand (CMD_WAKE_UP, 0x00);

  sendCommand (CMD_PLAY_FOLDER_FILE,
0x6002);

  delay(700);

}

```

```

else {

}

}

```

```

String decodeMP3Answer() {

  String decodedMP3Answer = "";

```

```

  decodedMP3Answer += sanswer();

  switch (ansbuf[3]) {

    case 0x3A:

      decodedMP3Answer += " -> Memory card
inserted.";

      break;

```

```

    case 0x3D:

      decodedMP3Answer += " -> Completed
play num " + String(ansbuf[6], DEC);

      break;

```

```

    case 0x40:

      decodedMP3Answer += " -> Error";

      break;

```

```

    case 0x41:

      decodedMP3Answer += " -> Data recived
correctly. ";

      break;

```

```

    case 0x42:

      decodedMP3Answer += " -> Status playing:
" + String(ansbuf[6], DEC);

      break;

```

```

    case 0x48:

```

```
    decodedMP3Answer += " -> File count: " +  
String(ansbuf[6], DEC);
```

```
    break;
```

```
case 0x4C:
```

```
    decodedMP3Answer += " -> Playing: " +  
String(ansbuf[6], DEC);
```

```
    break;
```

```
case 0x4E:
```

```
    decodedMP3Answer += " -> Folder file  
count: " + String(ansbuf[6], DEC);
```

```
    break;
```

```
case 0x4F:
```

```
    decodedMP3Answer += " -> Folder count:  
" + String(ansbuf[6], DEC);
```

```
    break;
```

```
}
```

```
return decodedMP3Answer;
```

```
}
```

```
/*  
*****  
*****/  
*/
```

```
/*Function: Send command to the MP3  
*/
```

```
/*Parameter: byte command  
*/
```

```
/*Parameter: byte dat1 parameter for the  
command          */
```

```
/*Parameter: byte dat2 parameter for the  
command          */
```

```
//void sendCommand(byte command){
```

```
// sendCommand(command, 0, 0);
```

```
//}
```

```
void sendCommand(int8_t command, int16_t  
dat){
```

```
    delay(20);
```

```
    Send_buf[0] = 0x7E; //
```

```
    Send_buf[1] = 0xFF; //
```

```
    Send_buf[2] = 0x06; // Len
```

```
    Send_buf[3] = command; //
```

```
    Send_buf[4] = 0x01; // 0x00 NO, 0x01  
feedback
```

```
    Send_buf[5] = (int8_t) (dat >> 8); // datah
```

```
    Send_buf[6] = (int8_t) (dat); // datah
```

```
    Send_buf[7] = 0xEF; //
```

```
    Serial.print("Sending: ");
```

```
    for (uint8_t i = 0; i < 8; i++)
```

```
    {
```



```

mp3.write(Send_buf[i] );
Serial.print(sbyte2hex(Send_buf[i]));
}
Serial.println();
}

/*****
*****
*****/

/*Function: sbyte2hex. Returns a byte data in
HEX format.          */

/*Parameter:- uint8_t b. Byte to convert to
HEX.                */

/*Return: String
*/

String sbyte2hex(uint8_t b)
{
String shex;

shex = "0X";

if (b < 16) shex += "0";

shex += String(b, HEX);

shex += " ";

return shex;
}

/*****
*****
*****/

/*Function: shex2int. Returns a int from an
HEX string.          */

/*Parameter: s. char *s to convert to HEX.
*/

/*Parameter: n. char *s' length.
*/

/*Return: int
*/

int shex2int(char *s, int n){

int r = 0;

for (int i=0; i<n; i++){

if(s[i]>='0' && s[i]<='9'){

r *= 16;

r +=s[i]-'0';

}else if(s[i]>='A' && s[i]<='F'){

r *= 16;

r += (s[i] - 'A') + 10;

}

}

return r;

}

```

```

/*****
*****
*****/

/*Function: sanswer. Returns a String answer
from mp3 UART module. */

/*Parameter:- uint8_t b. void.
*/

/*Return: String. If the answer is well
formatted answer. */

String sanswer(void)
{
    uint8_t i = 0;
    String mp3answer = "";

    // Get only 10 Bytes
    while (mp3.available() && (i < 10))
    {
        uint8_t b = mp3.read();
        ansbuf[i] = b;
        i++;

        mp3answer += sbyte2hex(b);
    }

    // if the answer format is correct.
    if ((ansbuf[0] == 0x7E) && (ansbuf[9] ==
0xEF))
    {
        return mp3answer;
    }
    return "???: " + mp3answer;
}

```



BIOGRAFI PENULIS

Muhammad Ridwan adalah nama dari penulis Tugas Akhir ini, penulis dilahirkan dari orang tua yang bernama Masrul dan ibunda Yurni sebagai anak kedua dari tiga persaudaraan. Penulis dilahirkan di Kota Pekanbaru, 02 Desember 1996, Kecamatan Pekanbaru, Kota Pekanbaru Riau. Penulis menempuh pendidikan dasar pertama dimulai dari SDN 004 Pandau Permai (*lulus tahun 2009*), melanjutkan ke SMPN 4 Bengkalis, (*lulus tahun 2012*) dan kemudian melanjutkan sekolah menengah atas/ MAN Bengkalis (*lulus tahun 2015*). Setelah selesai SMA akhirnya penulis melanjutkan pendidikan ke perguruan tinggi POLITEKNIK NEGERI BENGKALIS hingga selesai.

Akhir kata penulis mengucapkan puji syukur Alhamdulillah kehadiran Allah Subhanahu wata'ala kerana diberika kekuatan hingga mampu menyelesaikan penulisan Tugas Akhir ini dengan judul “**Timbangan Digital Berbasis Arduino Uno Dengan Output Suara**”. *Wassalamualaikum.*