

**RANCANG BANGUN *SMART RICE BOX*
DENGAN SISTEM PENAKAR BERAS BERDASARKAN
KEBUTUHAN PORSI MAKAN**

SKRIPSI

untuk memenuhi salah satu persyaratan
mencapai derajat Sarjana S1



Disusun oleh:

**Cahyo Setyawan Nino Arindra Karisma
13524076**

**JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
YOGYAKARTA**

2017

LEMBAR PENGESAHAN

LEMBAR PENGESAHAN

RANCANG BANGUN *SMART RICE BOX* DENGAN SISTEM PENAKAR BERAS BERDASARKAN KEBUTUHAN PORSI MAKAN

TUGAS AKHIR

Diajukan sebagai Salah Satu Syarat untuk Memperoleh
Gelar Sarjana Teknik
pada Program Studi Teknik Elektro
Fakultas Teknologi Industri
Universitas Islam Indonesia

Disusun oleh:

Cahyo Setyawan Nino A.K
13524076



Yogyakarta, 17 April 2018

Menyetujui,

Pembimbing I



Dwi Ana Ratna Wati S.T., M.Eng.
035240102

LEMBAR PENGESAHAN PENGUJI

LEMBAR PENGESAHAN PENGUJI

RANCANG BANGUN *SMART RICE BOX*

DENGAN SISTEM PENAKAR BERAS BERDASARKAN

KEBUTUHAN PORSI MAKAN

TUGAS AKHIR

Disusun Oleh :

Cahyo Setyawan Nino A.K

13524076

Telah Dipersembahkan di Depan Sidang Penguji Sebagai Salah Satu Syarat untuk
Memperoleh Gelar Sarjana Teknik Elektro

Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 15 Mei 2018

Tim Penguji,

Ketua

Dwi Ana Ratna Wati, S.T., M.Eng

Anggota I

Medilla Kusriyanto, ST, M.Eng

Anggota II

Yusuf Aziz Amrulloh, ST, M.Eng, Ph.D.

Mengetahui,

Ketua Jurusan Teknik Elektro

Universitas Islam Indonesia



Dr. Euis Hendra Setiawan, S.T., M.T.

LEMBAR PERNYATAAN

PERNYATAAN

Dengan ini Saya menyatakan bahwa:

1. Skripsi ini merupakan hasil karya sendiri, dan sepanjang pengetahuan Saya juga tidak mengandung karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka.
2. Informasi dan materi skripsi yang terkait hak milik, hak intelektual, dan paten merupakan milik bersama antara tiga pihak yaitu penulis, dosen pembimbing, dan Universitas Islam Indonesia. Dalam hal penggunaan informasi dan materi Skripsi terkait paten maka akan diskusikan lebih lanjut untuk mendapatkan persetujuan dari ketiga pihak tersebut diatas.

Yogyakarta, 17 April 2018



Cahyo Setyawan Nino A.K

KATA PENGANTAR

Segala puji dan syukur penulis haturkan kepada Allah SWT, karena atas limpahan rahmat, hidayah dan kasih sayang-Nya, penulis dapat menyelesaikan laporan tugas akhir ini sebagai salah satu syarat untuk memperoleh gelar sarjana. Teknik Elektro Pendidikan Strata Satu (S1) Fakultas Teknologi Industri, Universitas Islam Indonesia. Shalawat dan salam semoga tercurah kepada junjungan kita Muhammad SAW beserta keluarga dan para sahabat hingga akhir zaman.

Dalam penulisan laporan ini penulis banyak mendapatkan bantuan dan dukungan dari berbagai pihak, untuk itu penulis ingin mengucapkan banyak terima kasih kepada :

1. Allah SWT atas segala limpahan rahmat dan karunia-Nya.
2. Nabi Muhammad SAW sebagai suri tauladan dan panutan bagi umat manusia.
3. Orang tua tercinta Alm. Bpk Hanik Iswanto dan Ibu Kartini yang selalu mendoakan, menasehati, memotivasi dan membantu dalam segala sesuatunya.
4. Kakak saya Niko Arindra dan Novitasari yang selalu mendoakan dan mendukung dalam segala sesuatunya.
5. Bapak Dr.Eng Hendra Setiawan, S.T., M.T. selaku Ketua Jurusan Teknik Elektro Fakultas Teknologi Industri Universitas Islam Indonesia.
6. Ibu Dwi Ana Ratna Wati, ST., M.Eng. selaku dosen pembimbing tugas akhir saya yang telah membimbing dan mendampingi dengan sabar dan baik.
7. Segenap dosen Jurusan Teknik Elektro Fakultas Teknologi Industri Universitas Islam Indonesia yang telah membimbing serta menyalurkan ilmunya semasa dibangku perkuliahan maupun diluar bangku perkuliahan.
8. Saudara Fahriza Azwar, Ahmad Syarif, Dika Abadianto, Deni Pradana, Pampam, Ari, Mola, Syamsul, Jaya dan teman-teman Teknik Elektro angkatan 13 serta teman-teman Teknik Elektro UII pada umumnya yang bersedia menemani, membantu , menghilangkan stress dan memberikan saran hingga penelitian ini dapat diselesaikan.
9. Semua saudara/saudari di dalam UII maupun diluar UII yang memberikan pengaruh baik dan tidak dapat penulis sebutkan satu persatu. Terima kasih atas kerja sama dan dukungannya.

ARTI LAMBANG DAN SINGKATAN

- IOM : (*Institute of Medicine*)
- AKE : angka kecukupan energi
- AKK : angka kecukupan karbohidrat
- RB : *set point*, jumlah takaran beras
- C : konstanta input jumlah anak kecil
- D : konstanta input jumlah orang dewasa
- K : variabel porsi anak kecil
- B : variabel porsi orang dewasa
- e : rata-rata nilai *error*

ABSTRAK

Metode masyarakat dalam mengambil takaran beras untuk memenuhi kebutuhan makan dalam 1 hari masih menggunakan perkiraan. Metode tersebut dapat menyebabkan takaran yang tidak pas dan memberikan dampak yang kurang efektif dalam mengolah beras sebagai kebutuhan pangan sehari-hari. Tujuan penelitian ini adalah membuat suatu *smart rice box* dengan sistem penakar beras berdasarkan kebutuhan porsi makan dalam 1 hari untuk mengurangi pemborosan. Takaran porsi beras dikelompokkan menjadi 2 variabel yaitu porsi anak kecil dan orang dewasa. Metode penentuan algoritma takaran porsi beras ditentukan dengan melihat dan menghitung hasil *review* angka kecukupan gizi(AKG) oleh *IOM(Institute of Medicine)*. *Smart rice box* yang dibuat menggunakan *microcontroller* arduino uno. Proses input dilakukan melalui *keypad 4x4* dan LCD 16x2 sebagai *display monitoring*. Sistem buka tutup beras menggunakan *part acrylic* sebagai *valve* yang digerakkan oleh motor servo MG996R. Dan jumlah beras yang keluar akan diukur oleh sensor load cell dengan modul penguat HX711. Penimbangan oleh load cell bekerja cukup akurat dengan tingkat akurasi 99.7%. Saat *valve* menutup, motor servo bekerja cukup baik namun kurang efektif sebagai penggerak katup beras. Motor servo memiliki *delay* waktu berputar sebesar 0.55 detik saat kondisi katup menutup, yang mempengaruhi nilai *output* beras. *Smart rice box* juga dilengkapi fitur *monitoring* jumlah beras yang masih pada *rice box*. Data *monitoring* jumlah beras disimpan pada memori EEPROM.

Kata kunci : arduino uno, *IOM*, load cell, motor servo, *keypad*, LCD.

DAFTAR ISI

LEMBAR PENGESAHAN.....	i
LEMBAR PENGESAHAN PENGUJI	ii
LEMBAR PERNYATAAN	iii
KATA PENGANTAR.....	iv
ARTI LAMBANG DAN SINGKATAN	v
ABSTRAK	vi
DAFTAR ISI.....	vii
DAFTAR GAMBAR	ix
DAFTAR TABEL.....	x
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang Masalah	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah	2
1.4 Tujuan Penelitian	2
1.5 Manfaat Penelitian	2
BAB 2 TINJAUAN PUSTAKA	3
2.1 Studi Literatur	3
2.2 Tinjauan Teori.....	3
2.2.1 <i>Smart Rice Box</i>	3
2.2.2 Kecukupan Karbohidrat.....	4
BAB 3 METODOLOGI.....	5
3.1 Gambaran Umum Sistem <i>Smart Rice Box</i>	5
3.2 Perancangan <i>Hardware</i>	5
3.2.1 Perancangan Bagian Komponen <i>Smart Rice Box</i>	5

3.2.2 Perancangan <i>Part Rice Box</i>	7
3.3 Perancangan <i>Software</i>	8
3.3.1 Kalibrasi Load cell dengan HX711	10
3.3.2 Program Menu	10
3.3.3 Perancangan Algoritma Porsi Beras	11
3.3.4 Sistem <i>Valve</i> Beras	12
3.3.5 Sistem <i>Monitoring</i>	13
3.3.6 EEPROM	13
3.4 Teknik Pengujian	13
BAB 4 HASIL DAN PEMBAHASAN	14
4.1 Uji Akurasi Sensor Load cell.....	14
4.2 Pengujian Sistem Mekanik <i>Valve</i>	15
4.3 Sistem <i>Monitoring</i>	18
BAB 5 KESIMPULAN DAN SARAN	23
5.1 Kesimpulan	23
5.2 Saran	23
DAFTAR PUSTAKA	24
LAMPIRAN	25

DAFTAR GAMBAR

Gambar 3. 1 <i>Close loop control smart rice box</i>	5
Gambar 3. 2 Diagram <i>hardware</i>	6
Gambar 3. 3 Desain <i>smart rice box</i>	7
Gambar 3. 4 <i>Flow chart</i> program <i>smart rice box</i>	8
Gambar 4. 1 (a) Tampilan awal, (b) tekan (*) set input jumlah beras	18
Gambar 4. 2 (a) Tampilan <i>monitoring</i> , (b) petunjuk pengambilan beras.....	18
Gambar 4. 3 (a) Proses input kategori orang dewasa, (b) instruksi lanjut ke proses selanjutnya .	19
Gambar 4. 4 (a) Input kategori anak kecil, (b) simpan input dan proses beras keluar.....	19
Gambar 4. 5 (a) Informasi takaran <i>output</i> beras, (b) <i>monitoring</i> sisa beras pada <i>rice box</i>	19
Gambar 4. 6 Proses input isi ulang beras pada <i>rice box</i>	20
Gambar 4. 7 <i>Display</i> memori penyimpanan EEPROM.	20
Gambar 4. 8 <i>Smart rice box</i> tampak depan	21
Gambar 4. 9 <i>Smart rice box</i> tampak belakang	21
Gambar 4. 10 <i>Smart rice box</i> tampak samping	22

DAFTAR TABEL

Tabel 2. 1 Distribusi persentase energi makro dan angka kecukupan karbohidrat dan serat	4
Tabel 3. 1 Konfigurasi pin setiap komponen	6
Tabel 3. 2 Pengelompokan usia untuk variabel anak kecil dan orang dewasa.....	11
Tabel 3. 3 Hasil perhitungan algoritma takaran beras.....	12
Tabel 4. 1 Perbandingan hasil pengukuran load cell dengan timbangan digital	14
Tabel 4. 2 Hasil pengujian sistem, perbandingan nilai input dengan <i>output smart rice box</i>	15
Tabel 4. 3 Hasil pengujian sistem dengan pengurangan pada algoritma	16
Tabel 4. 4 Pengujian <i>smart rice box</i> setelah kalibrasi.....	17

BAB 1

PENDAHULUAN

1.1 Latar Belakang Masalah

Beras merupakan kebutuhan sehari-hari masyarakat Indonesia dan beras pun perlu di tempatkan pada wadah beras kusus yang bersih dan tertutup. *Rice box* merupakan suatu wadah yang digunakan untuk menyimpan beras dengan aman dan memudahkan untuk mengambil beras ketika diinginkan. *Rice box* yang beredar di masyarakat masih sederhana atau manual, dan belum mempunyai indikator apakah beras di dalamnya sudah habis atau belum.

Metode masyarakat dalam mengambil takaran beras masih menggunakan perkiraan untuk memenuhi kebutuhan makan dalam 1 hari. Orang-orang yang belum terbiasa mengolah beras untuk kebutuhan makan sehari-hari akan mengalami kesulitan untuk menakar beras yang pas. Hal ini dapat menyebabkan takaran yang tidak pas dan memberikan dampak yang kurang efektif dalam mengolah beras sebagai kebutuhan pangan sehari-hari.

Salah satu cara untuk membantu masyarakat dalam menakar beras untuk kebutuhan makan adalah dengan membuat *rice box* yang mampu menakar beras berdasarkan porsi makan secara otomatis. Melalui *smart rice box*, pengguna nantinya hanya perlu melakukan proses input berapa jumlah orang yang akan dimasakkan beras. Secara otomatis, nilai input akan diolah oleh sistem dan beras akan keluar dalam jumlah yang telah ditentukan. *Smart rice box* dilengkapi sistem *monitoring* untuk memantau dan memberikan informasi jumlah beras tersisa.

Setiap orang memiliki jumlah konsumsi nasi yang berbeda-beda tergantung dari kebutuhan tubuh. Kebutuhan tubuh itu juga dipengaruhi berdasarkan, usia, aktifitas fisik, jenis kelamin, berat dan tinggi badan, serta kesehatan atau penyakit. Melihat banyaknya faktor individu tersebut, maka metode yang diambil dalam penentuan takaran beras adalah menggunakan data angka kecukupan gizi (AKG) guna melihat standar kebutuhan konsumsi karbohidrat manusia dalam sehari. Metode ini diambil karena akan menjadi lebih terarah untuk menentukan kebutuhan porsi makanan yang ideal berdasarkan kesehatan atau ahli gizi. Meskipun nantinya dalam beberapa keluarga akan ada kurang kesesuaian antara hasil takaran *smart rice box* dengan konsumsi makan orang dikarenakan beberapa faktor individu, namun dengan adanya *smart rice box* mampu membantu dan memberikan pengetahuan tentang penakaran beras untuk kebutuhan makan berdasarkan AKG dan memudahkan penggunaan.

1.2 Rumusan Masalah

Bagaimana merancang sebuah *smart rice box* otomatis yang mampu memberikan takaran yang pas untuk kebutuhan makan dalam satu hari?

1.3 Batasan Masalah

1. Desain *smart rice box* dengan daya tampung maksimal 5 kg beras.
2. Interaksi pengendalian langsung pada *smart rice box*, tidak terintegrasi dengan *smartphone*, *remote* ataupun perangkat eksternal lainnya.
3. Pemilihan kategori takaran beras hanya ditujukan pada kategori porsi makan orang dewasa dan anak kecil untuk kebutuhan makan 1 hari.
4. Penentuan takaran porsi beras didasarkan pada data angka kecukupan gizi (AKG) pada jenis kelamin laki-laki.
5. Tidak memperhatikan perbedaan jenis beras.

1.4 Tujuan Penelitian

Mewujudkan *smart rice box* yang mampu menakar beras berdasarkan kebutuhan porsi makan dalam sehari untuk mengurangi pemborosan.

1.5 Manfaat Penelitian

1. Menambah wawasan mengenai pengoperasian piranti mikrokontroler dan membangun suatu inovasi alat yang layak digunakan.
2. Membantu masyarakat dalam menakar beras sesuai angka kecukupan gizi (AKG).
3. Mengurangi pemborosan dalam pengolahan beras dengan takaran yang pas.
4. Memberikan informasi jumlah beras di dalam *rice box* serta mengingatkan pengguna untuk mengisi beras ketika sisa beras hampir habis .
5. Sebagai bahan informasi dan referensi untuk pengembangan bagi penelitian berikutnya.

BAB 2

TINJAUAN PUSTAKA

2.1 Studi Literatur

Penelitian sebelumnya telah melakukan inovasi mengenai *rice box* berbasis mikrokontroler ATmega8535 yang dapat melakukan *monitoring* dan mengeluarkan beras secara otomatis dengan input berat beras yang di inginkan melalui *keypad* dan sensor berat menggunakan potensio geser. Mekanisme keluarnya beras (*valve*) menggunakan motor stepper sebagai penggerak. Pada penelitian ini berhasil menjalankan sistem *monitoring* dengan baik menggunakan LCD sebagai *display*. Namun sistem mekanik keluarnya beras kurang berjalan dengan baik dikarenakan motor stepper tidak sanggup menarik *belt* kaku yang diletakkan pada *acrylic* [1].

Pada penelitian lain yang dilakukan Yohana Susanthi dan Erwin Boenyamin Liem [2], yaitu merancang suatu sistem penimbangan otomatis untuk bahan kopi, gula, dan beras menggunakan mikrokontroler ATmega16. Perbedaan disini yaitu sistem mekanik yang digunakan menggunakan motor DC 12V (*Central Lock*) dan sensor berat menggunakan load cell. Hasilnya aktuator bekerja cukup baik pada objek gula dan beras. Sistem mekanik bekerja dengan baik, namun dipengaruhi oleh sifat setiap jenis objeknya.

Inovasi lain seperti [3], perancangan sebuah *rice box* dengan sistem *monitoring* dan mengeluarkan beras secara otomatis menggunakan android sebagai pengendalinya. Sistem ini menggunakan mikrokontroler ATmega16, sensor berat menggunakan load cell, sensor ultrasonik *HC-SR04* sebagai *monitoring* ada tidaknya beras di dalam *rice box*. Penampilan informasi menggunakan rangkaian LCD. Sedangkan aktuator yang digunakan untuk sistem mekanik menggunakan motor DC. Proses Input dan *monitoring* sistem dilakukan melalui android.

2.2 Tinjauan Teori

2.2.1 Smart Rice Box

Smart rice box merupakan suatu alat penyimpanan beras yang memiliki sistem otomatis untuk mengambil beras serta sistem *monitoring* jumlah beras di dalam wadah beras. Selain sebagai tempat penyimpanan beras, *smart rice box* berfungsi sebagai inovasi untuk membantu manusia dalam menakar beras untuk kebutuhan makan 1 hari, memberikan informasi berat takaran beras (gram) sebagai pengetahuan bagi pengguna, serta memberikan informasi tentang jumlah beras yang tersisa. Prinsip kerja *smart rice box* adalah, pengguna melakukan proses input jumlah

orang melalui *keypad*, nilai *set point* berupa takaran beras yang telah ditentukan, kemudian *valve* beras akan membuka dan beras keluar tertampung pada laci beras. Berat beras keluar akan ditimbang oleh sensor dan *valve* akan menutup ketika berat *output* beras sama dengan input.

2.2.2 Kecukupan Karbohidrat

Karbohidrat merupakan salah satu zat gizi makro yang berfungsi menyediakan energi bagi sel-sel tubuh. Kecukupan energi, kecukupan karbohidrat seseorang dipengaruhi oleh ukuran tubuh, berat badan, usia dan aktifitas fisik. Perbedaan faktor yang dimiliki tiap individu berimplikasi pada kecukupan karbohidrat yang berbeda-beda. Menurut data [4], salah satu cara untuk menghitung kebutuhan karbohidrat bagi setiap kelompok umur dan jenis kelamin yaitu didasarkan pada cara *by difference*. Total kecukupan energi dikurangi total energi dari kecukupan protein dan kecukupan lemak dengan syarat angka kecukupan energi, lemak, dan protein sudah diperoleh. Hasil perhitungan kecukupan karbohidrat berdasarkan cara *by difference* disajikan pada tabel 2.1.

Tabel 2. 1 Distribusi persentase energi makro dan angka kecukupan karbohidrat dan serat [4].

Umur	AKE 2012 (kkal)	%-Energi protein	%-Energi lemak	%-Energi Karbo	AKK 2012 (g)
0-5 bln	550	8	50	42	58
6-11 bln	725	10	45	45	82
1-3 th	1125	10	35	55	155
4-6 th	1600	10	35	55	220
7-9 th	1850	10	35	55	254
Laki-laki					
10-12 th	2100	15	30	55	280
13-15 th	2475	15	30	55	340
16-18 th	2675	15	30	55	368
19-29 th	2725	15	30	55	375
30-49 th	2625	15	25	60	394
50-64 th	2325	15	25	60	349
65-79 th	1900	10	25	65	309
80+ th	1525	10	25	65	248

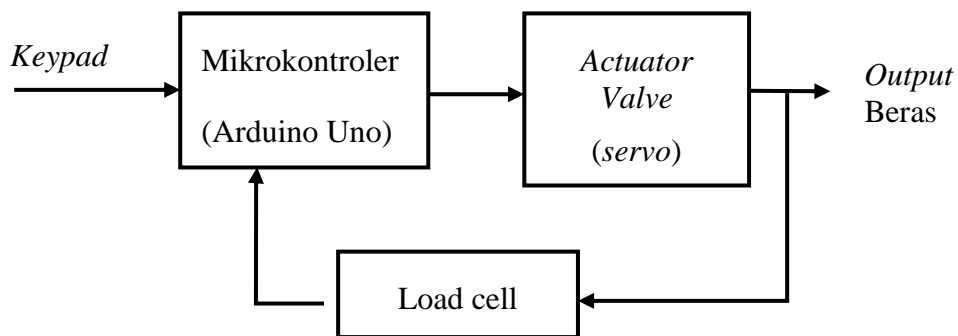
Penggunaan data angka kecukupan gizi (AKG) khususnya pada angka kecukupan karbohidrat(AKK), akan lebih merangkum informasi dan penentuan porsi beras untuk kebutuhan makan dalam 1 hari, karena informasi didasarkan pada data kesehatan. Dari data angka kecukupan karbohidrat diatas, akan di hitung nilai kecukupan karbohidrat (laki-laki) untuk kelompok umur kategori anak kecil dan orang dewasa. Setelah nilai kecukupan karbohidrat diperoleh akan dikonversikan menjadi jumlah beras (gram) yang dibutuhkan. Kandungan karbohidrat dalam 100 gram beras adalah sekitar 79 gram karbohidrat.

BAB 3

METODOLOGI

3.1 Gambaran Umum Sistem *Smart Rice Box*

Perancangan sistem *smart rice box* secara umum terbagi menjadi dua yaitu perancangan *hardware* dan perancangan *software*. Gambaran umum sistem disajikan pada gambar 3.1.



Gambar 3. 1 *Close loop control smart rice box*

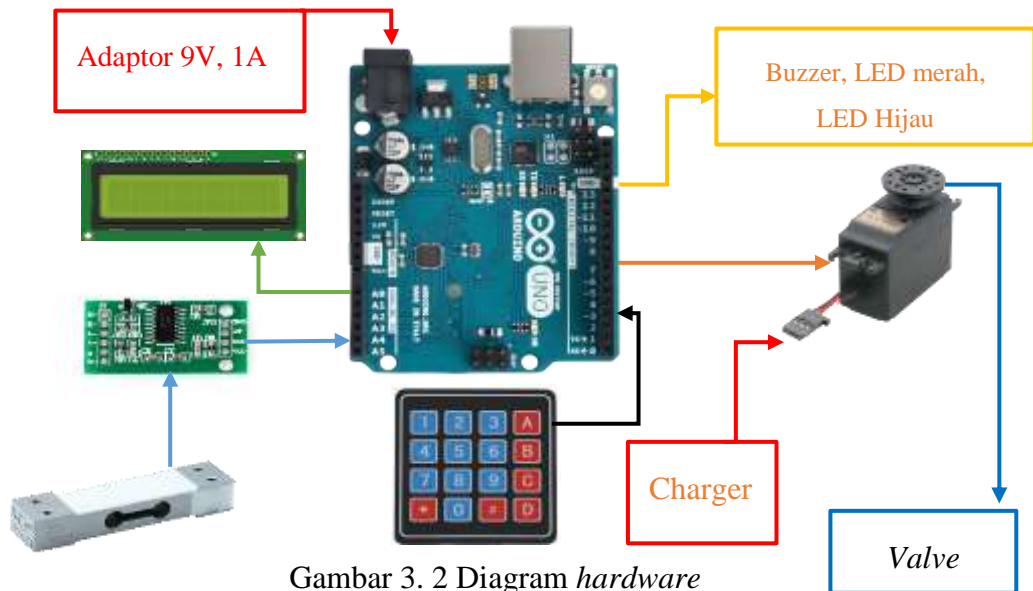
Perancangan *smart rice box* menggunakan sistem *closed loop*, dimana variabel yang diatur adalah jumlah takaran beras. Nilai beras yang keluar akan dibaca oleh sensor dan dibandingkan dengan *set point* nya. Nilai *set point* diperoleh dari algoritma porsi beras yang mengolah nilai input berupa jumlah orang sesuai kebutuhan.

3.2 Perancangan *Hardware*

Perancangan *hardware smart rice box* terbagi menjadi 2 yaitu: bagian komponen dan bagian *part* penyusun *rice box*.

3.2.1 Perancangan Bagian Komponen *Smart Rice Box*

Perancangan *interface* setiap komponen *smart rice box* disajikan pada gambar 3.3, dan tabel 3.1. *Supply* daya arduino menggunakan adaptor. Semua komponen dihubungkan pada pin – pin tertentu arduino dengan *supply* daya pada setiap komponennya berasal dari tegangan keluaran 5vdc arduino yang dirangkai secara paralel, kecuali motor servo menggunakan *supply* daya dari charger 5 volt , 1,5 ampere .



Gambar 3. 2 Diagram hardware

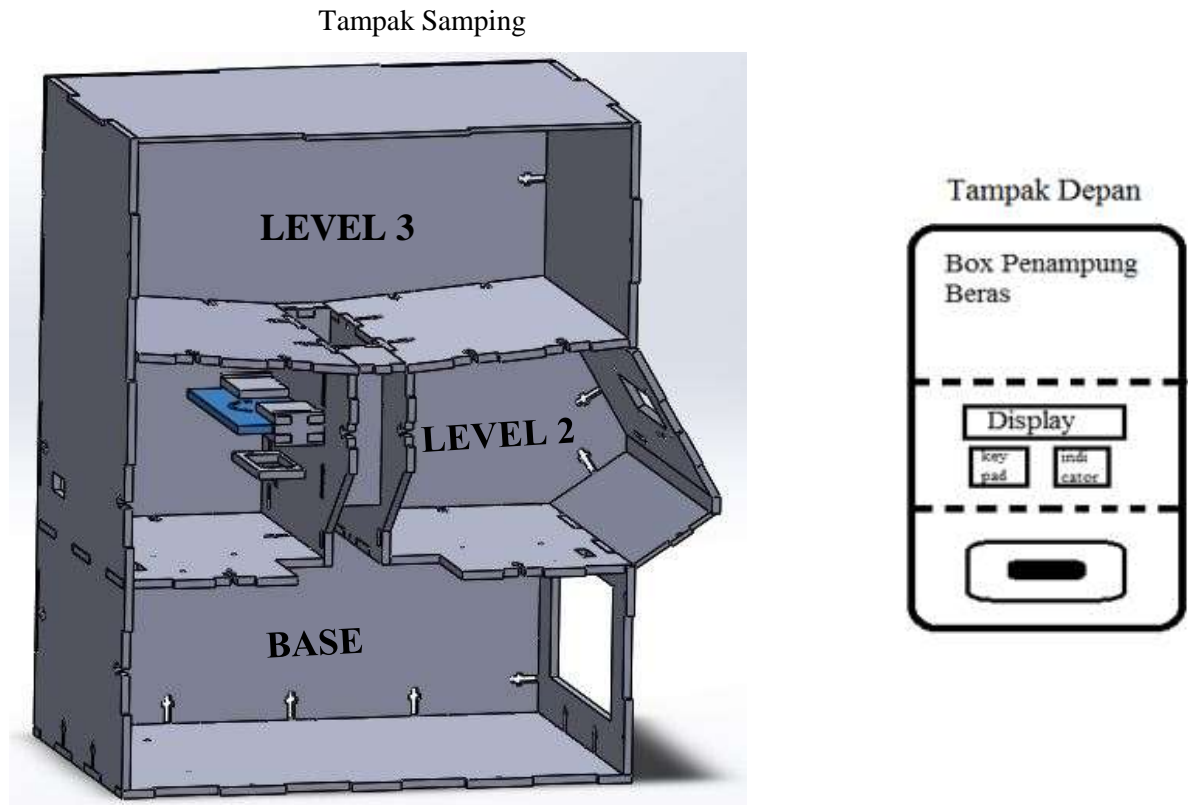
Tabel 3. 1 Konfigurasi pin setiap komponen

Konfigurasi pin	
Pin Komponen	Pin Arduino
HX711	
➤ DOUT	A2
➤ SCK	A3
I2C LCD	
➤ SDA	A4
➤ SCL	A5
KEYPAD 4x4	2, 3, 4, 5, 6, 7, 8, 9
MOTOR SERVO	10 (PWM)
Buzzer	13
LED merah, LED Hijau	11, 12
Supply Komponen	
➤ VDC	5VDC
➤ Ground	GND
➤ charger	

Hardware terdiri dari komponen input berupa keypad 4x4 berfungsi untuk proses input, dan load cell dengan modul penguat HX711 untuk mengukur berat beras yang keluar. Komponen output berupa LCD 16x2 terintegrasi I2C sebagai display, motor servo sebagai penggerak valve beras serta buzzer dan LED sebagai indicator monitoring beras. Semua komponen terhubung dengan microcontroller arduino uno.

3.2.2 Perancangan *Part Rice Box*

Perancangan desain wadah beras dari *cover* luar sampai *part-part* bagian dalam ditunjukkan pada gambar 3.3. Wadah beras dibuat dengan bahan *acrylic* 5mm.

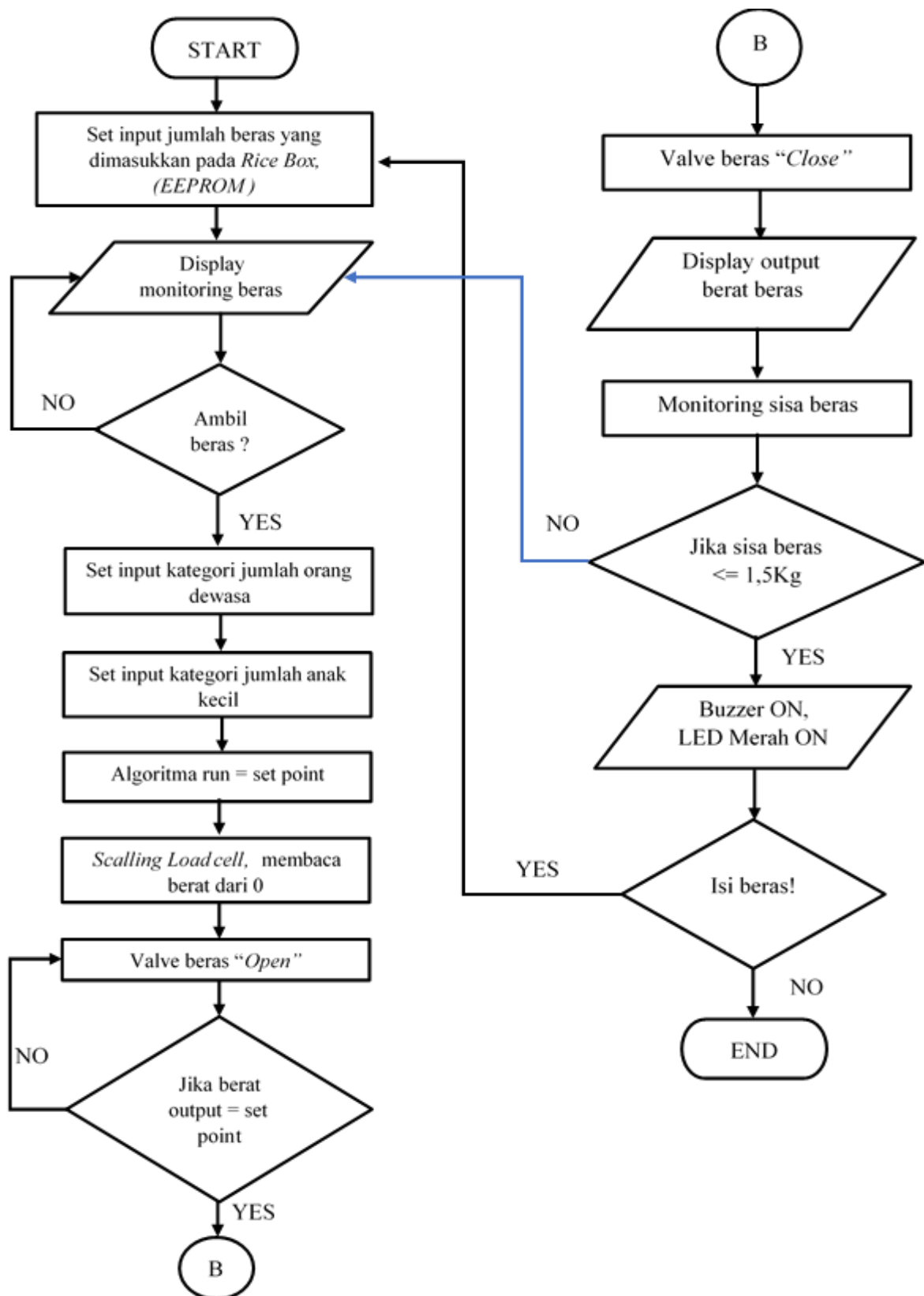


Gambar 3. 3 Desain *smart rice box*

Smart rice box yang dirancang terbagi menjadi 3 bagian, yaitu:

1. **Bagian base** : bagian paling bawah, di dalamnya terdiri dari laci beras sebagai tempat menampung *output* beras. Laci beras ditopang oleh dudukan laci yang terhubung dengan modul timbangan (load cell dan HX711) untuk menimbang jumlah *output* beras pada laci beras.
2. **Level 2** : merupakan bagian untuk sistem *smart rice box* berada. Di dalamnya terdapat arduino, komponen-komponen pendukung, kit mekanik *valve* beras serta corong jalur keluarnya beras, kemudian LCD, LED dan *Keypad* di bagian depannya.
3. **Level 3** : tempat paling atas, berfungsi sebagai tangki penampung beras dengan kapasitas maksimal yaitu 5 kg beras

3.3 Perancangan *Software*



Gambar 3. 4 *Flow chart* program smart rice box

Dari gambar 3.4, menunjukkan alur bagaimana proses kerja *smart rice box* berjalan, yaitu:

1. Pertama kali alat digunakan, *user* harus melakukan input manual jumlah beras(gram) yang dimasukkan ke dalam tangki *rice box* melalui *keypad* “tekan (*)”.
2. *Display monitoring*, menampilkan informasi jumlah beras pada tangki beras.
3. Ambil beras?, jika ingin ambil beras maka tekan “A”, akan masuk ke menu input.
4. *Set* input jumlah orang dewasa untuk memasukkan berapa jumlah orang dewasa yang ingin dimasakkan beras.
5. *Set* input jumlah anak kecil (tekan “B”) untuk memasukkan berapa jumlah anak kecil.
6. Setelah proses input selesai, maka nilai input akan tersimpan dan diproses di dalam algoritma sehingga menghasilkan jumlah beras (gram) yang harus keluar (*set point*).
7. *Scalling* load cell aktif, membaca berat dimulai dari 0 untuk mengukur berat beras yang akan keluar.
8. *Valve* beras membuka ketika proses input telah selesai.
9. Beras yang keluar akan tertampung pada laci beras yang terukur oleh sensor berat. Ketika berat *output* beras pada laci sama dengan nilai *set point* takaran beras, maka *valve* menutup.
10. Jumlah *output* beras terpenuhi dan akan ditampilkan nilainya pada LCD.
11. Kemudian sistem akan menghitung sisa beras pada tangki beras dengan cara jumlah beras pada tangki sebelumnya dikurangi jumlah beras yang keluar. Hasilnya akan ditampilkan pada LCD.
12. Kemudian program *monitoring* akan menghitung sisa beras pada tangki, dan akan melihat kondisi apabila jumlah beras pada tangki kurang dari 1500gr, maka *indicator* LED merah menyala dan alarm buzzer menyala. Dan ketika beras kurang dari 1500g *user* sebaiknya mengisi ulang beras dan melakukan *set* input jumlah isi ulang beras dengan menekan *keypad* “#”.

Dari rancangan pengendalian dan *flow chart* sistem *smart rice box*, dijadikan acuan dalam pembuatan program melalui *software* arduino. Pemrograman pertama melakukan kalibrasi load cell dengan *library* HX711 sehingga load cell mampu membaca beban dengan baik dalam satuan gram. Kemudian membuat program LCD untuk menampilkan hasil pembacaan sensor load cell. Lanjut ke pembuatan program menu pemilihan kategori input porsi beras orang dewasa dan anak kecil. Kemudian perancangan algoritma, sistem buka tutup *valve* dan sistem *monitoring*.

3.3.1 Kalibrasi Load cell dengan HX711

Cara kalibrasi load cell dengan HX711 supaya dapat berfungsi adalah:

1. Jalankan program pembacaan (*scalling*) HX711 dengan *library* HX711. Jalankan program dalam kondisi sensor load cell tanpa beban.
2. Buka serial monitor, ketika ada pembacaan sensor terbaca dan tampil nilainya, beri beban pada load cell. Dan akan tampil nilai *hexadecimal* pembacaan sensor.
3. Catat nilai pembacaan sensor dengan beban, kemudian hitung dengan persamaan :

$$\text{Faktor Kalibrasi (gram)} = \frac{\text{nilai pembacaan sensor(Hexa)}}{\text{berat beban uji (gram)}} \quad (3.1)$$

4. Kemudian nilai faktor kalibrasi yang diperoleh dimasukkan dalam fungsi *scale.set_scale(faktor kalibrasi)* pada *void setup*. *scale.set_scale()* berfungsi untuk menetapkan nilai *set* dari hasil faktor kalibrasi untuk digunakan sebagai pembagi dalam pembacaan beban. Sehingga pembacaan sensor terkalibrasi dan sudah dalam satuan gram.
5. Kemudian program pembacaan sensor terhadap beban dengan menggunakan fungsi *scale.get_units(10)*. *Get_units(10)* merupakan fungsi untuk menghitung rata-rata pembacaan sensor dalam 10 kali pembacaan. Nilai konstanta menunjukkan banyaknya jumlah pembacaan, semakin kecil nilainya maka proses *scalling* pembacaan berat akan semakin cepat.

3.3.2 Program Menu

Dalam pemrograman arduino untuk sistem *smart rice box*, berisi pemrograman untuk load cell, LCD 16x2, keypad 4x4, proses input jumlah beras pada tangki, menu untuk proses input pengambilan beras, sistem mekanik motor servo dan program untuk sistem *monitoring*. Program dibagi dalam beberapa *sketch* atau lembar, dengan satu *sketch* berfungsi sebagai main program. Di dalam setiap *sketch* memanggil tiap-tiap *function sketch* yang dibuat sesuai kebutuhan melalui fungsi “*void*” yang dibuat pada setiap *sketch*. Teknik ini digunakan karena sederhana dan praktis, membuat fungsi yang dapat dipanggil beberapa kali sesuai kebutuhan. Serta metode ini dapat mempermudah dalam proses analisa *error* program.

3.3.3 Perancangan Algoritma Porsi Beras

Tabel 3. 2 Pengelompokan usia untuk variabel anak kecil dan orang dewasa.

Laki - laki			Kebutuhan karbohidrat (gram)
Umur	%- Energi Karbohidrat	AKK (2012) gram	AKK x %Energi karbohidrat
4-6 th	55	220	121
7-9 th	55	254	140
10 - 12 th	55	289	154
13 - 15 th	55	340	187
16-18 th	55	368	202
19-29 th	55	375	206
30-49 th	60	394	236
50-64 th	60	349	209
65-79 th	60	309	201
80+ th	65	248	161

Pada tabel 3.2, baris 3 sampai 6 merupakan kelompok variabel anak kecil dan baris 7 sampai 12 adalah kelompok variabel orang dewasa. Maka dapat dihitung porsi berasnya:

Diketahui bahwa kandungan karbohidrat dalam 100 gram beras adalah sekitar 79 gram karbohidrat.

- Angka kecukupan karbohidrat anak kecil di golongan pada *range* umur 4th – 15th.

$$\text{Rata-rata AKK anak kecil} = \frac{121 + 140 + 154 + 187}{4} = 150,5 \text{ gram karbohidrat/ hari.}$$

$$\text{Kebutuhan porsi beras anak kecil} = \frac{100}{K} = \frac{79}{150,5} ; K = 190 \text{ gram beras/ hari.}$$

- Angka kecukupan karbohidrat orang dewasa di golongan pada *range* umur 16th – 80th.

$$\text{Rata-rata AKK orang dewasa} = \frac{202 + 206 + 236 + 209 + 201 + 161}{6} = 202,5 \text{ gram karbohidrat/}$$

hari.

$$\text{Kebutuhan porsi beras orang dewasa} = \frac{100gr}{B} = \frac{79}{202,5} ; B = 256 \text{ gram beras/ hari.}$$

Maka jumlah porsi beras untuk variabel anak kecil adalah $K = 190$ gram beras/ hari. Dan jumlah porsi beras untuk variabel orang dewasa adalah $B = 256$ gram beras/ hari. Sehingga algoritma takaran porsi beras adalah:

$$RB = (C*K)+(D*B) \quad (3.2)$$

Keterangan:

RB : *set point*, jumlah berat beras

C : konstanta input jumlah anak kecil

D : konstanta input jumlah orang dewasa

K : variabel porsi anak kecil

B : variabel porsi orang dewasa

Hasil perhitungan takaran beras untuk beberapa sampel pengambilan disajikan pada tabel 3.3.

Tabel 3. 3 Hasil perhitungan algoritma takaran beras

Input		Takaran Beras (g)	Input		Takaran Beras (g)	Input		Takaran Beras (g)
Dewasa	Anak kecil		Dewasa	Anak kecil		Dewasa	Anak kecil	
1	0	256	2	2	892	3	4	1528
1	1	446	2	3	1082	4	0	1024
1	2	636	2	4	1272	4	1	1214
1	3	826	3	0	768	4	2	1404
1	4	1016	3	1	958	4	3	1594
2	0	512	3	2	1148	4	4	1784
2	1	702	3	3	1338	5	0	1280

3.3.4 Sistem Valve Beras

Valve beras berada pada dinding corong beras. Prinsip kerja valve beras adalah pelat valve dari bahan *acrylic* 5 mm bergerak maju mundur dengan penggerak motor servo. Sistem penggeraknya berupa tuas motor servo terhubung pada lubang parabola pada pelat valve. Kondisi valve membuka saat posisi tuas servo 0^0 , dan kondisi valve menutup saat posisi tuas servo 90^0 . Dalam program, valve mendapat perintah untuk membuka, ketika proses input 2 variabel telah selesai dengan *finishing* menekan *enter* pada *keypad*. Kondisi valve menutup berdasarkan pembacaan dari load cell, ketika load cell membaca nilai *output* beras sama dengan input (*set point*) maka valve menutup.

3.3.5 Sistem *Monitoring*

Monitoring jumlah beras pada *rice box* dengan cara:

$$\text{monitoring} = \text{berat beras pada tangki (gr)} - \text{berat output beras(gr)} \quad (3.3)$$

Diketahui terlebih dahulu jumlah berat beras awal pada tangki beras dengan melakukan input manual berapa jumlah beras yang dimasukkan kedalam tangki beras. kemudian setiap ada pengambilan beras, nilai *output* beras akan dijadikan pengurang bagi jumlah beras pada tangki. Ketika jumlah beras pada tangki lebih dari 1500 g, LED hijau menyala. Jika sisa beras pada tangki beras kurang dari 1500 g, maka *buzzer* dan LED merah menyala. Begitu seterusnya proses akan berulang. Nilai jumlah beras pada tangki akan selalu ditampilkan pada LCD.

3.3.6 EEPROM

EEPROM adalah sebuah memori *non-volatile* atau memori yang tidak terhapus meskipun tidak dialiri arus listrik. EEPROM dapat dibaca dan ditulis dari program arduino [5]. EEPROM memiliki alamat dengan jumlah memori sebesar 1kb. Pada *smart rice box* EEPROM berfungsi menyimpan nilai jumlah beras pada tangki beras. Pemrograman dilakukan dengan menggunakan *library* EEPROM arduino. Untuk menyimpan data nilai jumlah beras ke dalam EEPROM, maka data harus dibagi kedalam beberapa alamat. [6] Pemrograman menggunakan fungsi EEPROM.put yaitu fungsi memasukkan nilai dalam EEPROM menggunakan *semantics variable*. EEPROM.get untuk membaca dan mengambil nilai yang telah disimpan pada EEPROM sebagai *float*.

3.4 Teknik Pengujian

Analisis sistem *smart rice box* dilakukan dengan cara ,melihat kinerja perangkat *smart rice box* seperti:

- Menganalisa keakuratan pengukuran berat beban dari sensor load cell dengan membandingkan hasil pengukuran dengan timbangan digital dari pabrik.
- Menguji sistem *smart rice box* meliputi kinerja sistem mekanik *valve*, apakah *valve* mampu membuka setelah set input selesai, dan *valve* menutup setelah jumlah porsi beras terpenuhi. Dan membandingkan hasil *output* beras dengan input (*set point*). Serta mengamati *display* dan sistem *monitoring smart rice box*.

BAB 4

HASIL DAN PEMBAHASAN

4.1 Uji Akurasi Sensor Load cell

Pengujian sensor load cell dengan membandingkan pengukuran berat dari load cell dibandingkan dengan timbangan digital pabrikan.

Tabel 4. 1 Perbandingan hasil pengukuran load cell dengan timbangan digital

Pengukuran Loadcell (g)	Timbangan Digital SF-400 (g)	error (g)	Pengukuran Loadcell (g)	Timbangan Digital SF-400 (g)	error (g)
77	78	-1	1419	1419	0
105	107	-2	1643	1642	1
173	174	-1	1989	1989	0
222	223	-1	2339	2339	0
329	330	-1	2699	2700	-1
397	397	0	2933	2933	0
450	450	0	3242	3239	3
698	698	0	3625	3621	4

Dari hasil pengujian sensor load cell, persentase nilai akurasi sensor adalah:

$$\text{persentase akurasi} = \frac{\text{hasil pengukuran Loadcell}}{\text{hasil pengukuran timbangan}} \times 100 \quad (4.1)$$

Persentase nilai *error* pembacaan sensor Load cell adalah:

$$\text{error} = \frac{(\text{hasil pengukuran Loadcell} - \text{hasil pengukuran timbangan})}{\text{hasil pengukuran timbangan}} \times 100 \quad (4.2)$$

Tabel 4.1 menunjukkan beberapa hasil pengujian dari pengukuran berat. sensor load cell cukup akurat dalam pembacaan berat. Rata-rata keakuratan sensor load cell dalam pengukuran berat sebesar 99,71%. Dan rata-rata persentase nilai *error* pembacaan sensor load cell sebesar 0.29%. Pembacaan load cell juga diuji pada objek mengalir seperti pasir dan beras. Hasilnya sensor load cell mampu membaca berat objek yang di kucurkan dalam wadah penampung dengan baik, dan tidak terjadi *error*. Load cell mampu mengukur secara linear dari berat beban yang terus bertambah.

4.2 Pengujian Sistem Mekanik Valve

Pengujian untuk melihat apakah sistem mekanik *valve* mampu bekerja dengan baik, serta membandingkan nilai *output* dengan input.

Tabel 4. 2 Hasil pengujian sistem, perbandingan nilai input dengan *output smart rice box*

Uji Coba <i>Smart Rice box</i>			Percobaan 1		Percobaan 2		Percobaan 3	
Input		set point (g)	<i>output 1</i> (g)	$\Delta error$ (g)	<i>output 2</i> (g)	$\Delta error$ (g)	<i>output 3</i> (g)	$\Delta error$ (g)
Dewasa	Anak kecil							
1	0	256	348	92	333	77	327	71
1	1	446	537	91	524	78	519	73
1	2	636	729	93	716	80	704	68
1	3	826	903	77	902	76	900	74
2	0	512	594	82	585	73	590	78
2	1	702	776	74	776	74	766	64
2	2	892	973	81	965	73	956	64
2	3	1082	1162	80	1145	63	1151	69
3	0	768	848	80	844	76	846	78
3	1	958	1045	87	1045	87	1035	77
3	2	1148	1245	97	1216	68	1215	67
RATA-RATA NILAI <i>ERROR</i>				84.9091		75		71.1818

Dari hasil uji coba, sistem buka tutup *valve* beras bekerja cukup baik. Apabila dilihat pada tabel 4.2, terdapat *error* dimana nilai *output* beras lebih besar dari nilai input (*set point*), hal ini dikarenakan motor servo sebagai penggerak *valve* memiliki *delay* waktu untuk berputar. Pada beberapa kali uji coba, rata-rata *delay* waktu putaran motor servo saat kondisi “*open*” sebesar 0.5 detik, dan *delay* waktu saat kondisi “*close*” sebesar 0.55 detik. Akibat dari *delay* putaran motor servo ini adalah berat *output* beras menjadi bertambah. Sehingga saat load cell membaca nilai *output* beras sama dengan input, servo sudah menerima sinyal menutup katup, namun ada *delay* sebesar 0.55 detik baru *valve* benar-benar tertutup. Sehingga beras masih mengucur pada *delay* 0.55 detik dan menyebabkan *output* beras bertambah. Pada percobaan, diketahui bahwa desain corong beras pada tangki kurang miring sehingga ketika sisa beras sedikit, beras yang berada di ujung kurang bisa turun ke lubang corong.

Dari 3 hasil percobaan, rata-rata nilai *error output* beras sebesar 77 gram beras berlebih. Perbaikan nilai *error* pada *output* beras dilakukan dengan cara, nilai rata-rata *error output* beras dimasukkan ke dalam algoritma takaran beras sebagai pengurang hasil perhitungan takaran beras.

Persamaan algoritma menjadi:

$$RB = (C*K)+(D*B) - e \tag{4.3}$$

Keterangan:

- RB : *set point*, jumlah berat beras
- C : konstanta input jumlah anak kecil
- D : konstanta input jumlah orang dewasa
- K : variabel porsi anak kecil
- B : variabel porsi orang dewasa
- e : rata-rata nilai *error*.

Tabel 4.3 merupakan hasil pengujian kembali *smart rice box* dengan perubahan algoritma (persamaan 4.3), pada beberapa pengurangan nilai algoritma.

Tabel 4. 3 Hasil pengujian sistem dengan pengurangan pada algoritma

Kalibrasi <i>Smart Rice box</i>			e = 77 g		e = 80 g		e = 83g	
Input		set point (g)	output (g)	$\Delta error$ (g)	output (g)	$\Delta error$ (g)	output (g)	$\Delta error$ (g)
Dewasa	Anak kecil							
1	0	256	270	14	259	3	263	7
1	1	446	454	8	453	7	448	2
1	2	636	644	8	644	8	643	7
1	3	826	834	8	833	7	829	3
2	0	512	528	16	517	5	509	-3
2	1	702	711	9	707	5	704	2
2	2	892	908	16	901	9	901	9
2	3	1082	1085	3	1091	9	1084	2
3	0	768	782	14	777	9	763	-5
3	1	958	964	6	964	6	961	3
3	2	1148	1150	2	1150	2	1147	-1
RATA-RATA NILAI <i>ERROR</i>				9.45455		6.36364		2.36364

Setelah dilakukan perbaikan untuk mengurangi besarnya nilai *error*, hasil dari pengurangan pada algoritma sebesar 83 g memberikan hasil yang lebih baik dengan kelebihan atau kekurangan pada *output* beras tidak begitu besar. Sehingga dari proses kalibrasi di tetapkan pengurang pada algoritma sebesar 83 g dan akan diuji kembali untuk melihat tingkat presisi alat.

Tabel 4. 4 Pengujian *smart rice box* setelah kalibrasi

Uji Coba <i>Smart Rice Box</i>		Percobaan 1			Percobaan 2		Percobaan 3	
Input		set point (g)	<i>output</i> (g)	$\Delta error$ (g)	<i>output</i> (g)	$\Delta error$ (g)	<i>output</i> (g)	$\Delta error$ (g)
Dewasa	Anak kecil							
1	1	446	455	9	452	6	447	1
1	2	636	640	4	647	11	639	3
1	3	826	829	3	824	-2	841	15
2	0	512	517	5	514	2	518	6
2	1	702	708	6	708	6	698	-4
2	2	892	896	4	892	0	892	0
2	3	1082	1085	3	1090	8	1084	2
3	0	768	775	7	773	5	765	-3
3	1	958	957	-1	964	6	965	7
3	2	1148	1146	-2	1144	-4	1150	2
RATA-RATA NILAI <i>ERROR</i>				3.8		3.8		2.9

Lanjutan Tabel 4.4.

Uji Coba <i>Smart Rice Box</i>		Percobaan 4			Percobaan 5	
Input		set point (g)	<i>output</i> (g)	$\Delta error$ (g)	<i>output</i> (g)	$\Delta error$ (g)
Dewasa	Anak kecil					
1	1	446	445	-1	448	2
1	2	636	637	1	648	12
1	3	826	829	3	824	-2
2	0	512	524	12	515	3
2	1	702	706	4	704	2
2	2	892	893	1	892	0
2	3	1082	1080	-2	1084	2
3	0	768	771	3	766	-2
3	1	958	957	-1	967	9
3	2	1148	1152	4	1145	-3
RATA-RATA NILAI <i>ERROR</i>				2.4		2.3

Tabel 4.4 merupakan hasil beberapa kali percobaan setelah dilakukan kalibrasi, yaitu pengurangan $e = 83$ g pada algoritma. Dapat dilihat bahwa hasil *output* beras kurang konstan, hal ini dipengaruhi oleh beberapa faktor seperti tekanan beras pada tangki, dimana tangki berisi penuh beras akan memiliki tekanan lebih besar dibanding tangki berisi sedikit beras, sehingga akan mempengaruhi intensitas kucuran beras. Selain itu, faktor *scalling* pembacaan load cell dalam pengukuran memiliki pengaruh penting pada penentuan waktu penutupan katup beras. Karena kecepatan dan ketepatan *scalling* load cell yang menentukan kapan servo harus menutup. Serta faktor kinerja dari aktuator *valve* dan kurang sempurnanya desain sistem mekanik *valve*, dimana

pelat *valve* terlalu tebal (5mm) dan pengukuran desain yang kurang akurat untuk mengubah gerak rotasi motor servo menjadi gerak translasi *valve* beras, yaitu posisi lubang parabola dan motor servo kurang bergeser sedikit, sehingga di salah satu sisi katup kurang bisa menutup rapat. Namun *valve* beras tetap dapat beroperasi cukup baik. Dan dilihat dari hasil percobaan bahwa lama kelamaan *smart rice box* bekerja lebih baik dengan nilai *error* yang semakin berkurang (Percobaan 3,4,5).

4.3 Sistem *Monitoring*

Hasil dan analisa dari proses pengambilan beras yang ditampilkan pada LCD. Apakah sistem *monitoring* beras dapat berjalan dengan baik. Apakah sistem mampu menghitung sisa beras pada tangki beras dari setiap pengambilan beras, dan menampilkan pada LCD.



(a)



(b)

Gambar 4. 1 (a) Tampilan awal, (b) tekan (*) set input jumlah beras

Pada gambar 4.1 menunjukkan proses *set* input jumlah beras yang dimasukkan pada tangki beras. Pengguna harus melakukan *set* input jumlah beras saat pertama kali menggunakan alat.



(a)



(b)

Gambar 4. 2 (a) Tampilan *monitoring*, (b) petunjuk pengambilan beras

Setelah selesai *set* input beras, maka akan ditampilkan di LCD seperti gambar 4.2 yang juga merupakan tampilan *monitoring* beras. Pada tampilan *monitoring* juga terdapat petunjuk untuk mengambil beras dengan menekan tombol “A” untuk memilih kategori.



(a)



(b)

Gambar 4. 3 (a) Proses input kategori orang dewasa, (b) instruksi lanjut ke proses selanjutnya

Pada gambar 4.3 menunjukkan proses untuk mengambil beras yang diawali dengan input kategori orang dewasa. Setelah selesai, tekan *enter* dan muncul petunjuk untuk melakukan *set* input kategori anak kecil dengan menekan tombol "B".



(a)



(b)

Gambar 4. 4 (a) Input kategori anak kecil, (b) simpan input dan proses beras keluar

Gambar 4.4 menunjukkan proses input kategori anak kecil. Setelah selesai tekan *enter* maka nilai input akan tersimpan dan ditampilkan pada LCD.



(a)



(b)

Gambar 4. 5 (a) Informasi takaran *output* beras, (b) *monitoring* sisa beras pada *rice box*

Setelah proses pengambilan beras selesai, LCD akan menampilkan informasi jumlah takaran beras yang diambil, seperti gambar 4.5. Kemudian sistem akan kembali menampilkan monitoring sisa beras pada *rice box*.



Gambar 4. 6 Proses input isi ulang beras pada *rice box*.

Ketika sisa beras pada *rice box* tinggal sedikit dan pengguna ingin melakukan isi ulang. Maka pengguna juga harus melakukan set input jumlah beras sesuai berat beras yang dimasukkan ke dalam tangki beras. Proses *set* input isi ulang beras seperti pada gambar 4.6, dengan menekan tombol “#”, masukkan nilai input melalui *keypad* kemudian tekan *enter*.



Gambar 4. 7 *Display* memori penyimpanan EEPROM.

Gambar 4.7 merupakan hasil pengujian program penyimpanan menggunakan EEPROM. Pengujian dilakukan dengan mencabut *supply* daya yang masuk ke *smart rice box*, kemudian menghubungkan kembali alat dengan sumber tegangan. Hasilnya *monitoring* jumlah beras berhasil tersimpan pada EEPROM dan dapat ditampilkan pada LCD.

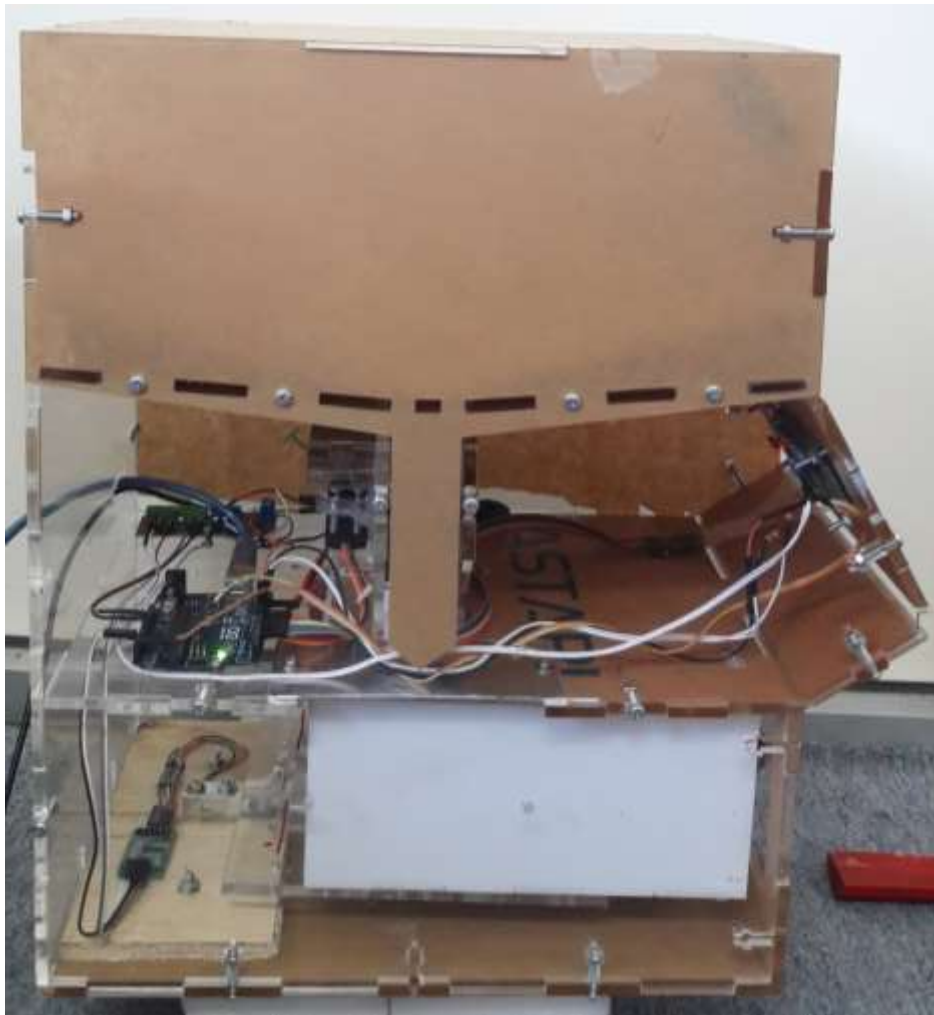
Pada percobaan, *display* dan *monitoring* dapat bekerja dengan baik. *smart rice box* mampu beroperasi dengan baik sesuai konsep *flow chart* yang telah dibuat dan setiap proses dapat ditampilkan pada LCD. *Monitoring* sisa beras bekerja dengan baik, setiap proses mengambil beras selesai, jumlah takaran *output* beras menjadi pengurang pada jumlah beras dalam *rice box*. Dan *indicator* LED hijau mampu menyala ketika jumlah beras lebih dari 1500 g. LED merah akan menyala ketika jumlah beras kurang dari 1500 g diikuti bunyi beep buzzer sebanyak 13 kali sebagai *reminder*. Karna LCD yang dipakai LCD 16x2 penampilan informasi hanya terbatas.



Gambar 4. 8 *Smart rice box* tampak depan



Gambar 4. 9 *Smart rice box* tampak belakang



Gambar 4. 10 *Smart rice box* tampak samping

Gambar 4.8, 4.9, dan 4.10 merupakan bentuk *hardware smart rice box*. *Rice box* mampu menyimpan beras dengan kapasitas maksimal 5 kg beras. Kapasitas laci beras dapat menampung maksimal 1.5 kg beras. Ketika pengguna mengambil beras dengan jumlah melebihi kapasitas maksimum laci beras, maka akan muncul pemberitahuan “*over load*” pada LCD dan beras tidak akan keluar. Pada kasus ini pengguna harus mengambil ulang beras. Bila pengguna ingin mengambil beras lebih dari 1.5 kg maka harus dilakukan 2 kali pengambilan.

BAB 5

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan percobaan, *smart rice box* mampu beroperasi dengan baik serta mampu membantu dan memberikan informasi dalam menakar beras untuk kebutuhan makan dalam 1 hari. Kapasitas maksimal laci beras sebesar 1.5 kg. Akurasi sensor load cell dalam membaca berat sebesar 99,7%. Nilai *output* jumlah beras untuk sama dengan *set point* dipengaruhi beberapa faktor, yaitu: kinerja motor servo sebagai penggerak valve memiliki *delay* waktu berputar sebesar 0,55 detik saat menutup, tekanan beras pada tangki beras, proses *scalling* load cell dalam pembacaan berat untuk objek mengalir. Kelemahan *smart rice box* yang dirancang adalah tingkat kemiringan corong beras yang kurang tajam, penggunaan motor servo sebagai aktuator masih kurang efektif dengan desain sistem mekanik yang kurang sempurna, menyebabkan proses menutup *valve* kurang rapat. Serta ketepatan hasil takaran beras untuk memenuhi kebutuhan keluarga belum bisa dikatakan 100% tepat dan efisien, karena hanya menggunakan 2 variabel porsi beras (dewasa dan anak kecil) yang didasarkan pada jenis kelamin laki-laki. Dan proses pengujian diperlukan beberapa sampel orang atau keluarga dan waktu yang lama untuk dapat dianalisa.

5.2 Saran

Saran untuk penelitian lebih lanjut:

- Daya tampung beras yang lebih besar.
- Penambahan kategori variabel porsi beras serta jenis kelamin.
- Pengembangan algoritma penentuan takaran beras.
- Sistem buka tutup katup beras yang lebih baik.
- Sistem informasi jumlah sisa beras pada *rice box* dibuat otomatis.
- Pengujian takaran porsi beras secara langsung kebeberapa keluarga dalam kurun waktu tertentu.

DAFTAR PUSTAKA

- [1] M. S. Gultom, A. N. Jati, F. I. Terapan, and U. Telkom, "Sistem Monitoring Beras pada Rice Box Berbasis Mikrokontroler," Bandung, 2012.
- [2] Y. Susanthi, B. Liem, J. T. Elektro, and U. K. Maranatha, "Sistem Penimbangan Otomatis Menggunakan Mikrokontroler ATmega16," *Electr. Eng. J.*, vol. 1, no. 1, pp. 41–52, 2010.
- [3] J. W. Akbar, "Rancang Bangun Alat Penakar Beras Otomatis Menggunakan Android Berbasis Mikrokontroler," Politeknik Negeri Sriwijaya, Palembang, 2016.
- [4] V. Napitupulu, H. Riyadi, and Hardiansyah, "Kecukupan Energi, Protein, Lemak dan Karbohidrat," *ResearchGate*, no. 1, p. 27, 2013.
- [5] B. Earl, "Memories of an Arduino," *adafruit Learn. Syst.*, vol. 1, no. 1, pp. 1–25, 2018.
- [6] Arduino Inc., "Arduino - EEPROM," *Arduino*, 2018. [Online]. Available: <http://arduino.cc/es/Reference/EEPROM>.

LAMPIRAN

Lampiran 1: Listing Program (Main Program)

```
#include <EEPROM.h>

#include <HX711.h>

#include <Keypad.h>

#include <Wire.h>

#include <LiquidCrystal_I2C.h>

#include <Servo.h>

int i=0;

int a=7;

int C;

int D;

int K;

int B;

float Str;

int nyetock;

int RB;

int porsi;

int siber;

long weight;

int w=0;

byte index=0;

byte indexx=0;

byte indexS=0;

byte indexS2=0;
```

```

char num1[16];

char num2[16];

char numS[16];

char numS2[16];

char keyp;

int y =0;

int z =0;

int s =0;

int v=0;

const int buzzer = 13;

float beras;

int eeAddress = 0;

struct MyObject {

    float field1;

    byte field2;

    char name[10];

};

LiquidCrystal_I2C lcd(0x3F, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE);

const byte ROWS = 4;

const byte COLS = 4;

char hexaKeys[ROWS][COLS] = {

    {'1','2','3','+'},

    {'4','5','6','-'},

    {'7','8','9','?'},

    {'<','0','>','='}

};

```

```

byte rowPins[ROWS] = {2, 3, 4, 5}; //connect to the row pinouts of the keypad

byte colPins[COLS] = {6, 7, 8, 9}; //connect to the column pinouts of the keypad

Keypad customKeypad = Keypad( makeKeymap(hexaKeys), rowPins, colPins, ROWS,
COLS);

Servo servoMain;

float seed = 212.13;

HX711 scale(A2, A3);

void setup() {

Serial.begin(9600);

scale.set_scale(seed);

scale.tare();

lcd.begin(16,2);

servoMain.attach(10); // servo on digital pin 11

servoMain.write(90);

pinMode(buzzer, OUTPUT);

pinMode(11, OUTPUT);

pinMode(12, OUTPUT);

get_storage();

lcd.clear();

lcd.setCursor (0, 0);

lcd.print("beras:");

lcd.setCursor (7, 0);

lcd.print(beras);

lcd.print("g ");

}

```

```

void put_storage() {

float f = 1234; //Variable to store in EEPROM.

int eeAddress = 0; //Location we want the data to be put.

EEPROM.put(eeAddress, f);

Serial.println("Written float data type!");

MyObject customVar = { Str };

eeAddress += sizeof(float); //Move address to the next byte after float 'f'.

EEPROM.put(eeAddress, customVar);

}

void get_storage(){

float f = 0; //Variable to store data read from EEPROM.

int eeAddress = 0;

Serial.print("Read float from EEPROM: ")

EEPROM.get(eeAddress, f); //get alamat data eeprom

Serial.println(f, 3);

secondTest();

}

void secondTest() {

int eeAddress = sizeof(float); //Move address to the next byte after float 'f'.

MyObject customVar; //Variable to store custom object read from EEPROM.

EEPROM.get(eeAddress, customVar);

beras = customVar.field1;

Serial.println("Read custom object from EEPROM: ");

Serial.println(customVar.field1);

Serial.println(customVar.field2);

}

```

```

void loop() {

  z=0;

  y=0;

  v=0;

  s=0;

  keyp=customKeypad.getKey();

  monitoring();

}

```

Lampiran 2: Listing Program, *Sketch* timbangan (Load cell)

```

void Load_cell() {

  scale.set_scale(seed);

  weight = scale.get_units(1);

  if (weight <=0) { weight = 0; }

  Serial.print("Weight:\t");

  Serial.println(weight, 1);

}

```

Lampiran 3: Listing Program, *Sketch* Servo (Valve)

```

void servo() {

  algoritma();

  get_storage();

  scale.tare();

  Str=beras;

  while(v>0) {

    servoMain.attach(10);

    if (RB>1528) {

      servoMain.write(115);

```



```

delay(1000);

lcd.clear();

  lcd.setCursor (0, 0);

  lcd.print("Maaf, Over Load");

  lcd.setCursor (0, 1);

  lcd.print("Mhon Ambil ulang");

  index=0;

  indexx=0;

  num1[index]='\0';

  num2[indexx]='\0';

  a=7;

  delay(3000);

  v=0;

}

```

```

else if (RB<=1528) {

  Load_cell();

  servoMain.write(0);

  if ( weight >= porsi) {

  servoMain.write(90);

  delay(500);

  siber= Str - RB;

  Str=siber;

  delay(1000);

  lcd.clear();

  lcd.setCursor (0, 0);

  lcd.print("Takaran=");

  lcd.setCursor (9, 0);

  lcd.print(RB);

```

```

lcd.print(" g");

delay(3000);

v=0;

}

}

}

display();

}

```

Lampiran 4: Listing Program, *Sketch* Algoritma

```

void algoritma() {

    D=D;

    C=C;

    K=190;

    B=256;

    RB = ((D*B)+(C*K));

    porsi = ((D*B)+(C*K))-83;

}

```

Lampiran 5: Listing Program, *Sketch* Program Menu

```

void dewasa() {

while(z<1) {

    keyp=customKeypad.getKey(); // CHOOSE ZONE

    Serial.print(keyp);

    if(keyp =='+'){

        lcd.clear();

        lcd.setCursor (0, 0);

        lcd.print("input kategori");

        lcd.setCursor (0, 1);

```

```

lcd.print("Dewasa");

delay(1000);

lcd.clear();

lcd.setCursor (0, 0);

lcd.print("Dewasa");

lcd.setCursor (0, 1);

lcd.print("input:");
}

if (keyp>='0'&&keyp<='9'){

lcd.setCursor(a,1);

lcd.print(keyp);

a=a+1;

num1[index++]=keyp;

num1[index]='\0';

}

if (keyp=='='){

Serial.println(num1);

D = atoi(num1);

a=7;

delay(1000);

lcd.clear();

lcd.setCursor (0, 0);

lcd.print("Pilih anak kecil ");

lcd.setCursor (0, 1);

lcd.print("Tekan down");

delay(500);

child();

z=1;

```

```

    }
cancel();
}
}
void child() {
    while(y<1){
        keyp=customKeypad.getKey();
        if(keyp =='-'){
            lcd.clear();
            lcd.setCursor (0, 0);
            lcd.print("input kategori");
            lcd.setCursor (0, 1);
            lcd.print("anak kecil");
            delay(1000);
            lcd.clear();
            lcd.setCursor (0, 0);
            lcd.print("anak kecil");
            lcd.setCursor (0, 1);
            lcd.print("input:");
        }
        if (keyp>='0'&&keyp<='9'){
            lcd.setCursor(a,1);
            lcd.print(keyp);
            a=a+1;
            num2[indexx++]=keyp;
            num2[indexx]='\0';
        }

        if (keyp=='='){

```

```

Serial.println(num2);

C = atoi(num2);

a=7;

delay(500);

lcd.clear();

lcd.setCursor (0, 0);

lcd.print("selesai");

delay(500);

lcd.setCursor (0, 1);

lcd.print("nilai tersimpan");

delay(1000);

lcd.clear();

lcd.setCursor (0, 0);

lcd.print("input dewasa=");

lcd.setCursor (13, 0);

lcd.print(D);

lcd.setCursor (0, 1);

lcd.print("input child =");

lcd.setCursor (13, 1);

lcd.print(C);

delay(1000);

    algoritma();

v=1;

y=1;

servo();

}

cancel();

}

}

```

Lampiran 6: Listing Program, *Sketch* Input Jumlah Beras

```
void set_storage() {
  get_storage();
  while(s<1)
  {
  keyp=customKeypad.getKey();
  if(keyp == '<'){
    lcd.clear();
    lcd.setCursor (0, 0);
    lcd.print("isi jumlah beras");
    lcd.setCursor (0, 1);
    lcd.print("input:");
  }
  if (keyp>='0'&&keyp<='9'){
  lcd.setCursor(a,1);
  lcd.print(keyp);
  a=a+1;
  numS[indexS++]=keyp;
  numS[indexS]='\0';
  }
  if (keyp=='='){
  Serial.println(numS);
  Str = atoi(numS);
  Str= beras + Str;
  a=7;
  delay(1000);
  lcd.clear();
```

```

lcd.setCursor (0, 0);

lcd.print("input:");

lcd.setCursor (7, 0);

lcd.print(Str);

lcd.print(" g");

    display();

s=1;

}

cancel();

}

}

```

Lampiran 7: Listing Program, Isi Ulang Beras

```

void set_storage2() {

    get_storage();

    while(s<1)

    {

keyp=customKeypad.getKey();

if(keyp == '>'){

    lcd.clear();

    lcd.setCursor (0, 0);

    lcd.print("isi ulang beras");

    delay(500);

    lcd.setCursor (0, 1);

    lcd.print("input:");

}

    if (keyp>='0'&&keyp<='9'){

lcd.setCursor(a,1);

```

```

lcd.print(keyp);

a=a+1;

numS2[indexS2++]=keyp;

numS2[indexS2]='\0';

}

if (keyp=='='){

Serial.println(numS2);

nyetock = atoi(numS2);

nyetock = beras + nyetock;

a=7;

delay(1000);

Str = nyetock;

lcd.clear();

delay(1000);

displey();

indexS2=0;

s=1;

}

cancel();

}

}

```

Lampiran 8: Listing Program, *Sketch Display*

```

void displey() {

  put_storage();

  delay(1000);

  lcd.clear();

  lcd.setCursor(0,0);

```



```

lcd.print("beras:");
delay(500);
lcd.setCursor(7,0);
lcd.print(Str);
lcd.print(" gr");
delay(500);
lcd.setCursor(0,1);
lcd.print("ambil beras =(A)");
delay(1000);
  index=0;
  indexx=0;
  indexS=0;
  indexS2=0;
  num1[index]='\0';
  num2[indexx]='\0';
  numS[indexS]='\0';
  numS2[indexS2]='\0';

  if(Str > 1500) {
    digitalWrite(12, HIGH);
    digitalWrite(11, LOW);
  }
  else if (Str < 1500) {
    digitalWrite(11, HIGH);
    digitalWrite(12, LOW);
    lcd.setCursor(0,1);
    lcd.print("Segera isi Beras");

```

```

for(i=0; i < 3; i++){
    tone(buzzer, 1000); // send 1KHz sound signal
    delay(1000);
    noTone(buzzer);
    delay(1000);
}
}
}

```

Lampiran 9: Listing Program, *switch menu*

```

void monitoring() {

    keyp=customKeypad.getKey();
    if(keyp) // Check for a valid key.
    {
switch(keyp)
    {
        case '<':
            lcd.clear();
            lcd.setCursor(0,0);
            lcd.print("input beras (gr)");
            delay(500);
            lcd.setCursor(0,1);
            lcd.print("tekan ( * )");
            set_storage();

            break;

```

```

    case '+':
lcd.clear();
lcd.setCursor(0,0);
lcd.print("pilih kategori");
delay(500);
lcd.setCursor(0,1);
lcd.print("tekan ( A )");
dewasa();
break;

case '>':
lcd.clear();
    lcd.setCursor(0,0);
lcd.print("isi ulang beras");
delay(500);
lcd.setCursor(0,1);
lcd.print("tekan ( # )");
set_storage2();
    break;
}
}
}

```

Lampiran 10: Listing Program, *Cancel* sistem

```

void cancel() {
    if (keyp=='?'){
        index=0;
        indexx=0;

```

```

indexS=0;

indexS2=0;

num1[index]='\0';

num2[indexx]='\0';

numS[indexS]='\0';

numS2[indexS2]='\0';

a=7;

display();

Serial.print(keyp);

z=1;

y=1;

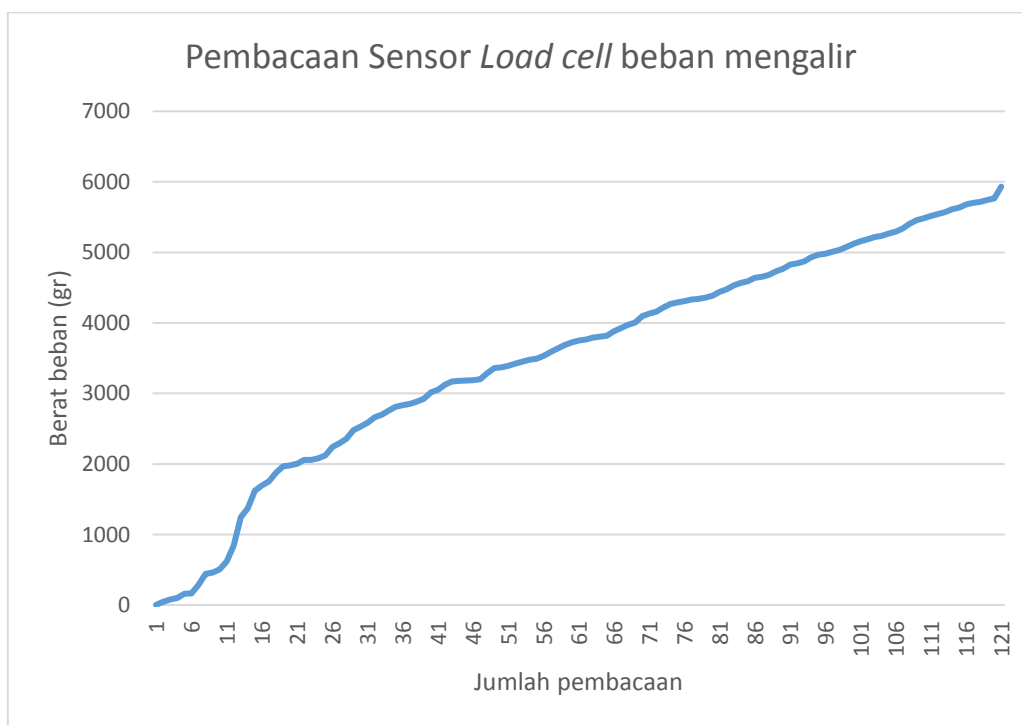
s=1;

}

}

```

Lampiran 11: Pembacaan Sensor Load cell beban mengalir



Lampiran 12: Hasil Kalibrasi *Smart Rice box*, (Algoritma -83gr)

