

**PROTOTIPE VENDING MACHINE MINUMAN KALENG
BERBASIS ARDUINO MEGA 2560**



Riandy Heryanda


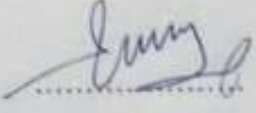
5215102636

**Skripsi ini Disusun Sebagai Salah Satu Persyaratan untuk
Memperoleh Gelar Sarjana Pendidikan**

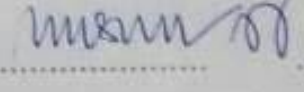

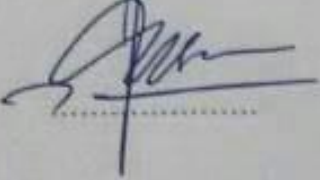
**PENDIDIKAN TEKNIK ELEKTRONIKA
JURUSAN TEKNIK ELEKTRO FAKULTAS TEKNIK
UNIVERSITAS NEGERI JAKARTA**

2017

HALAMAN PENGESAHAN

NAMA DOSEN	TANDA TANGAN	TANGGAL
<u>Drs. Jusuf Bintoro, MT.</u> (Dosen Pembimbing I)		28-8-2017
<u>Svufrijal, ST., MT.</u> (Dosen Pembimbing II)		28-8-2017

PENGESAHAN PANITIA UJIAN SKRIPSI

NAMA DOSEN	TANDA TANGAN	TANGGAL
<u>Drs. Wisnu Djatmiko, MT.</u> (Ketua Penguji)		28-8-2017
<u>Dr. Efri Sandi, MT.</u> (Sekretaris Penguji)		28-8-2017
<u>Dr. Muhammad Yusro, MT.</u> (Dosen Ahli)		28-8-2017

Tanggal Lulus: 18 Agustus 2017

HALAMAN PERNYATAAN

Dengan ini saya menyatakan bahwa:

1. Karya tulis skripsi saya ini adalah asli dan belum pernah diajukan untuk mendapatkan gelar akademik sarjana, baik di Universitas Negeri Jakarta maupun di perguruan tinggi lain.
2. Karya tulis ini adalah murni gagasan, rumusan dan penelitian saya sendiri dengan arahan dosen pembimbing.
3. Dalam karya tulis ini tidak terdapat karya atau pendapat yang telah ditulis atau dipublikasikan orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan dicantumkan dalam daftar pustaka.
4. Pernyataan ini saya buat dengan sesungguhnya dan apabila di kemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka saya bersedia menerima sanksi akademik berupa pencabutan gelar yang telah diperoleh karena karya tulis ini, serta sanksi lainnya sesuai dengan norma yang berlaku di Universitas Negeri Jakarta.

Jakarta, 18 Agustus 2017

Yang membuat pernyataan



Riandy Heryanda

5215102636

ABSTRAK

Riandy Heryanda, *Prototipe Vending Machine Minuman Kaleng Berbasis Arduino Mega 2560*. Jakarta, Program Studi Pendidikan Teknik Elektronika, Fakultas Teknik, Universitas Negeri Jakarta, 2017. Dosen Pembimbing: Drs. Jusuf Bintoro, MT. dan Syufrijal, ST., MT.

Tujuan penelitian ini adalah untuk membuat prototipe *vending machine* untuk menjual minuman kaleng menggunakan alat pembayaran *e-money*. Metode yang digunakan dalam penelitian ini yaitu rekayasa teknik yang meliputi analisis kebutuhan, perancangan, pembuatan, pengujian, dan implementasi sistem. Penelitian dilakukan di Universitas Negeri Jakarta yang dimulai pada November 2016 hingga Juni 2017.

Prototipe *vending machine* minuman kaleng yang dibuat pada penelitian ini berbasis Arduino Mega 2560 serta menggunakan *E-money* sebagai alat transaksi pembelian minuman kaleng. Prototipe mesin ini menyediakan 9 macam jenis minuman kaleng ukuran 330 ml yang setiap jenisnya tersimpan pada *storage*. Setiap *storage* dapat menyimpan maksimal 3 minuman kaleng, jadi total maksimal minuman kaleng yang dapat di simpan dalam *vending machine* ini berjumlah 27 minuman kaleng.

Hasil penelitian ini menunjukkan prototipe *vending machine* minuman kaleng berbasis Arduino Mega 2560 dapat berfungsi sesuai rancangan untuk mengambil/menyediakan minuman kaleng kepada pembeli. Sistem *E-Money* yang telah dibuat dapat berfungsi untuk alat bayar saat transaksi pembelian minuman kaleng pada unit *vending machine*.

Kata kunci: Prototipe, *Vending machine*, Minuman Kaleng, Arduino, MySQL Workbench, QT.

ABSTRACT

Riandy Heryanda, *Prototype of Canned Drink Vending Machine Based on Arduino Mega 2560*. Jakarta, Program Studi Pendidikan Teknik Elektronika, Fakultas Teknik, Universitas Negeri Jakarta, 2017. Dosen Pembimbing: Drs. Jusuf Bintoro, MT. dan Syufrijal, ST., MT.

The purpose of this research is to create a prototype vending machine to sell canned drinks using e-money payments. The methods used in this study is engineering research which includes needs analysis, designing, manufacturing, testing, and implementation of the system. Research conducted at the State University of Jakarta beginning in November 2016-June 2017.

Prototype of canned drink vending machine based on Arduino Mega 2560 and using E-money as a means of purchasing drinks cans. The prototype of this machine provides 9 kinds of beverage cans the size of 330 ml each type stored on the storage. Each storage can store a maximum of 3 cans, so total a maximum of can that can be stored in the vending machine was 27.

The results of this study demonstrate a prototype of canned drinks vending machine based on Arduino Mega 2560 may function according the draft to take/drink cans provide to the buyer. The system of E-Money that has been made can serve to pay when purchasing a beverage cans in vending machine unit.

Keywords: Prototype, vending machine, canned drinks, Arduino, MySQL Workbench, QT.

KATA PENGANTAR

Puji dan syukur kehadiran Allah SWT Yang Maha Pengasih dan Maha Penyayang. Atas rahmat dan hidayah-Nya, sehingga peneliti dapat menyelesaikan skripsi dengan judul **PROTOTIPE VENDING MACHINE MINUMAN KALENG BERBASIS ARDUINO MEGA 2560**. Skripsi ini disusun sebagai salah satu syarat dalam memperoleh gelar Sarjana Pendidikan pada Program Studi Pendidikan Teknik Elektronika, Fakultas Teknik, Universitas Negeri Jakarta.

Selama penyusunan skripsi, banyak bantuan yang peneliti terima dari berbagai pihak. Oleh karena itu, dalam kesempatan ini peneliti mengucapkan terima kasih kepada:

1. Bapak Drs. Pitoyo Yuliatmojo, MT , selaku Koordinator Program Studi Pendidikan Teknik Elektronika, Jurusan Teknik Elektro, Fakultas Teknik, Universitas Negeri Jakarta.
2. Dr. Moch. Sukardjo, M.Pd., selaku Pembimbing Akademik
3. Bapak Drs. Jusuf Bintoro, MT, selaku Dosen Pembimbing I.
4. Bapak Syufrijal, ST., MT, selaku Dosen Pembimbing II
5. Bapak Tjatja Surianda, dan Ibu Dede Jumaenah selaku kedua orang tua peneliti. Terima kasih sudah membesarkan, mendidik, merawat, dan mendoakan saya sehingga dapat menempuh pendidikan sejak usia dini sampai kuliah dan menyelesaikan skripsi ini.

Peneliti menyadari bahwa skripsi ini masih jauh dari kesempurnaan, karenanya peneliti mohon maaf apabila terdapat kekurangan kesalahan baik dari isi maupun tulisan. Akhir kata peneliti berharap semoga skripsi ini bermanfaat bagi yang membacanya.

Jakarta, 11 Agustus 2017

DAFTAR ISI

	Halaman
HALAMAN PENGESAHAN.....	ii
HALAMAN PERNYATAAN	iii
ABSTRAK	iv
ABSTRACT.....	v
KATA PENGANTAR	vi
DAFTAR ISI.....	vii
DAFTAR TABEL.....	x
DAFTAR GAMBAR	xi
DAFTAR LAMPIRAN.....	xiv
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang.....	1
1.2. Identifikasi Masalah.....	3
1.3. Pembatasan Masalah.....	4
1.4. Rumusan Masalah.....	5
1.5. Tujuan Penelitian	5
1.6. Manfaat Penelitian	5
BAB II TINJAUAN PUSTAKA.....	7
2.1. Prototipe.....	7
2.2. Vending machine	7
2.2.1. Konsep Vending machine	7
2.2.2. Sejarah Vending Machine (VM).....	8
2.2.3. Vending machine di Indonesia	9
2.3. Arduino	10
2.3.1. Definisi Arduino	10
2.3.2. Arduino Mega 2560	11

2.3.3.	Arduino Software IDE	12
2.4.	E-Money	13
2.5.	MySQL Workbench.....	14
2.6.	RFID	15
2.7.	Text LCD 16x2	18
2.8.	I2C (Inter Integrated Circuit).....	19
2.9.	Motor DC	20
2.10.	Motor Stepper	22
2.11.	Sensor.....	23
2.11.1.	Definisi Sensor.....	23
2.11.2.	Jenis-jenis Sensor.....	24
2.11.3.	Sensor Mekanik	24
2.11.4.	Sensor Cahaya.....	25
2.12.	A4988 Driver Motor Stepper	26
BAB III METODOLOGI PENELITIAN.....		29
3.1.	Tempat dan Waktu Penelitian.....	29
3.2.	Alat dan Bahan Penelitian.....	29
3.2.1.	Alat.....	29
3.2.2.	Bahan	30
3.3.	Diagram Alir Penelitian	30
3.3.1.	Tahap Analisis	31
3.3.2.	Tahap Perancangan	32
3.4.	Rancangan Mekanik	33
3.5.	Rancangan Elektronik.....	34
3.6.	Rancangan Siklus Kerja Vending Machine dan Program Arduino IDE	47
3.7.	Rancangan Sistem E-Money (database dan aplikasi administrator)...	51
3.8.	Teknik dan Prosedur Pengumpulan Data.....	52
BAB IV HASIL PENELITIAN DAN PEMBAHASAN.....		53

4.1.	Deskripsi Hasil Penelitian.....	53
4.1.1.	Hasil Rancangan Mekanik	54
4.1.2.	Hasil Rancangan Elektronik	58
4.1.3.	Hasil Rancangan Software.....	59
4.1.4.	Hasil Pembuatan Sistem Kerja Alat.....	63
4.2.	Analisis Data Penelitian	65
4.2.1.	Pengujian Actuator	65
4.2.2.	Pengujian Sensor Cahaya.....	70
4.2.3.	Pengujian E-Money	71
4.2.4.	Waktu Pelayanan Vending machine Minuman Kaleng	72
4.3.	Pembahasan	72
4.4.	Aplikasi Hasil Penelitian	72
BAB V KESIMPULAN DAN SARAN.....		73
5.1.	Kesimpulan	73
5.2.	Saran	73
DAFTAR PUSTAKA		74
LAMPIRAN.....		76
DAFTAR RIWAYAT HIDUP.....		123

DAFTAR TABEL

	Halaman
Tabel 2.1 Spesifikasi Arduino Mega 2560.....	12
Tabel 2.2 Spesifikasi A4988	27
Tabel 2.3 Pilihan Step Mode pada driver A4988.....	28
Tabel 3.1 Daftar Penggunaan Pin Arduino Mega 2560 Untuk Sensor Cahaya	39
Tabel 3.2 Hubungan Rangkaian Tombol dengan Arduino Mega 2560	42
Tabel 3.3 Hubungan Rangkaian Limit Switch dengan Arduino Mega 2560.....	45
Tabel 4.1 Hubungan Pin Rangkaian Elektronik.....	58
Tabel 4.2 Pengujian Waktu Tempuh Sumbu X	67
Tabel 4.3 Pengujian Waktu Tempuh Sumbu Y	68
Tabel 4.4 Koordinat minuman kaleng berdasarkan step pergerakan motor stepper	69
Tabel 4.5 Hasil Pengukuran Tegangan Actuator Sumbu Z.....	69
Tabel 4.6 Hasil Pengukuran Tegangan Actuator Pintu Tempat Pengambilan Minuman Kaleng.....	70
Tabel 4.7 Pengukuran Tegangan Fotodiode.....	71
Tabel 4.8 Pengukuran Tegangan Keluaran Rangkaian Sensor Cahaya	71
Tabel 4.9 Hasil Pengujian Waktu Pelayanan Pengambilan Minuman Kaleng	72

DAFTAR GAMBAR

	Halaman
Gambar 2.1. mesin self-acting untuk pengiriman pos serta perangko penerimaan di Inggris dan mesin jual permen karet berbentuk badut di Amerika	9
Gambar 2.2 Vending machine yang ada di Jepang	9
Gambar 2.3 Logo Arduino	10
Gambar 2.4 Papan Arduino Mega 2560.....	11
Gambar 2.5 Screenshot Tampilan Arduino IDE pada PC.....	13
Gambar 2.6 Logo MySQL Workbench.....	14
Gambar 2.7 Ilustrasi Interaksi Komponen RFID	16
Gambar 2.8 Penggunaan RFID reader dan contactless smart card digunakan secara praktis.....	17
Gambar 2.9 Text LCD 16x2.....	18
Gambar 2.10 Contoh Instalasi Text LCD pada Papan Arduino.....	19
Gambar 2.11 I2C LCD Backpack yang sudah terpasang pada Text LCD.....	20
Gambar 2.12 Aturan Tangan Kiri Fleming untuk Hubungan Arah Arus ,Medan Magnet, dan Gaya	21
Gambar 2.13 Motor DC	22
Gambar 2.14 Motor Stepper Sederhana Dimana Setiap Langkah Menggerakkan Rotor Sebesar 60 derajat	23
Gambar 2.15 Blok Fungsional Sensor/Transduser.....	24
Gambar 2.16 Limit Switch, Sensor Mekanik Pendeteksi Batas Gerakan Suatu Objek.....	25
Gambar 2.17 Simbol dan Bentuk Fisik Fotodiode.....	25
Gambar 2.18 Simbol dan Bentuk Fisik Fotodiode.....	26
Gambar 2.19 Simbol dan Bentuk Fisik Fototransistor.....	26
Gambar 2.20 Pinout A4988 dan Wiring Diagram Minimum Sistem Driver	27
Gambar 3.1. Diagram Alir Penelitian	31
Gambar 3.2. Diagram Blok Sistem Vending machine Minuman Kaleng Berbasis Arduino Mega 2560	33

Gambar 3.3 Rancangan mekanik vending machine.....	34
Gambar 3.4 Board arduino.....	35
Gambar 3.5 Motor Stepper Nema 17.....	35
Gambar 3.6 Modul A4988.....	36
Gambar 3.7 Skematik Pengendali Motor Stepper.....	36
Gambar 3.8 Skematik Actuator Sumbu X dan Sumbu Y (Motor DC).....	37
Gambar 3.9 Skematik Sensor Cahaya (pada box tempat pengambilan minuman).....	38
Gambar 3.10 Modul RFID Reader MFRC522 dan MIFARE Cards.....	39
Gambar 3.11 Skematik RFID Reader.....	40
Gambar 3.12 Modul RFID USB Plug and Play.....	41
Gambar 3.13 Instalasi Modul SL500-USB (RFID reader plug and play) ke Notebook PC.....	41
Gambar 3.14 Rancangan Skematik Tombol Pilihan Minuman Kaleng.....	42
Gambar 3.15 Skematik LCD 16x2 I2C ke Arduino Mega 2560.....	43
Gambar 3.16 Rancangan Panel Interface.....	44
Gambar 3.17 Limit Switch.....	44
Gambar 3.18 Rancangan Skematik Limit Switch Sebagai Pendeteksi Batasan Pergerakan Actuator.....	45
Gambar 3.19 Skematik Buzzer.....	46
Gambar 3.20 Flowchart Program.....	50
Gambar 3.21 Rancangan EER Diagram database E-Money pada MySQL Workbench.....	51
Gambar 3.22 Rancangan tampilan interface program Emoney_Database.exe yang dibuat menggunakan Qt Creator.....	52
Gambar 4.1 Mekanik Actuator Sumbu X.....	55
Gambar 4.2 Mekanik Actuator Sumbu Y.....	55
Gambar 4.3 Mekanik Actuator Sumbu Z.....	56
Gambar 4.4 Box tempat pengambilan minuman.....	56
Gambar 4.5 Mekanik Pintu Tempat Pengambilan Minuman Kaleng.....	57
Gambar 4.6 Mekanik Conveyor Storage (tempat persediaan minuman).....	57

Gambar 4.7 Tampilan program Sketch Arduino yang telah dibuat di Software Arduino IDE.....	60
Gambar 4.8 Tabel Admin pada database E-Money	61
Gambar 4.9 Database pada Table Card.....	62
Gambar 4.10 Database Pada Table Log.....	62
Gambar 4.11 Tombol Power Vending Machine (kiri) dan Tombol Power Notebook PC (kanan).....	63
Gambar 4.12 Aplikasi Emoney_database.exe.....	63
Gambar 4.13 Hubungkan Aplikasi dengan Vending Machine	64

DAFTAR LAMPIRAN

Lampiran 1. Dokumentasi.....	77
Lampiran 2. Gambar Teknik.....	80
Lampiran 3. Data-Data Pengukuran.....	81
Lampiran 4. <i>Source Code</i>	85

BAB I

PENDAHULUAN

1.1. Latar Belakang

Teknologi semakin berkembang dengan cepat dan dinamis. Perkembangan teknologi ini juga berkaitan dengan perubahan gaya hidup manusia yang semakin canggih dan inovatif, terutama dalam bidang teknologi yang dibuat serba otomatis untuk mempermudah pekerjaan manusia. Salah satu hasil kreasi teknologi di bidang perdagangan adalah suatu alat jual otomatis yang disebut *vending machine*. Sejak kemunculan *vending machine* dipasaran, maka hal ini menjadi suatu cara baru pada sistem penjualan. Di berbagai negara, sudah ada produsen minuman kemasan yang menggunakan jasa mesin untuk menjual produknya

Di Benua Amerika, Eropa, dan Asia, *vending machine* sudah dipasarkan dan digunakan secara umum, hal ini terlihat dari hak paten yang di keluarkan oleh *United States Patent and Trademark Office (USPTO)* dan *European Patent Office (EPO)*. Ada beberapa klaim yang dikeluarkan oleh lembaga-lembaga tersebut terkait *vending machine*. Di Negara Jepang bahkan sangat mudah menjumpai *Vending machine* di perkotaan bahkan pedesaan.

Peneliti memilih *vending machine* sebagai alat untuk dijadikan bahan penelitian dalam pembuatan “Prototipe *Vending machine* Berbasis Arduino Mega 2560”. Desain yang akan dirancang terinspirasi dari *vending machine* yang sudah ada sebelumnya. *Vending machine* rancangan baru ini diharapkan bisa bekerja dengan optimal dan bisa memperbaiki hal yang belum terselesaikan pada produk-produk sebelumnya.

Secara umum, sistem kerja *vending machine* berjenis mesin penjual minuman yang ada saat ini yaitu, yang pertama pembeli memilih minuman dengan menekan tombol pilihan, yang kedua pembeli diminta melakukan transaksi pembelian, yang ketiga *actuator* menggerakkan minuman agar berpindah ke lokasi tempat pengambilan minuman, lalu yang keempat pelanggan bisa mengambil minuman di dalam wadah dan proses pembelian dinyatakan selesai.

Pada proses tersebut terdapat sebuah transaksi pembelian yang harus dibayarkan oleh pembeli (pengguna mesin), pada proses ini pembeli biasanya diminta memasukkan sejumlah uang tunai ke dalam unit mesin transaksi. Peneliti berencana mengganti alat bayar dengan menggantikan uang tunai dengan *E-Money*.

Beberapa desain mekanik *vending machine* yang telah ada, untuk setiap pilihan minuman akan ada *actuator* masing-masing di setiap wadah tempat penyimpanan, yakni spiral yang diputar menggunakan motor stepper. Ada juga desain yang menggunakan *actuator* yang bergerak menuju wadah penyimpanan minuman berdasarkan koordinat sumbu X dan Y. Peneliti akan menggunakan desain mekanik dengan menggunakan *actuator* yang bergerak berdasarkan sumbu X, Y, Z, dan 1 *actuator* penggerak *conveyor* wadah tempat menyimpan minuman. Hal yang berbeda dari desain-desain pendahulunya adalah peneliti membuat wadah tempat penyimpanan minuman dengan bentuk *conveyor* kecil. Penggunaan *actuator* sumbu X, Y, Z, penggerak *conveyor* dan *conveyor* tempat penyimpanan minuman bertujuan untuk meminimalisir jumlah *actuator* dan agar minuman berpindah tempat dengan lebih aman karena *actuator* bergeser tanpa

menjatuhkan minuman, sehingga produk minuman tetap terjaga dan terhindar dari kerusakan. Prototipe *vending machine* ini dirancang untuk ditempatkan di suatu lingkup kecil dimana ada 1 administrator penjual kartu *E-Money*, lalu calon pembeli minuman kaleng bisa membeli saldo *E-Money* pada administrator. Setelah memiliki sejumlah saldo tertentu maka selanjutnya pembeli minuman dapat terus membeli di *vending machine* selama saldo saldo masih mencukupi.

Peneliti ingin mewujudkan *vending machine* minuman kaleng berbasis Arduino Mega 2560 yang merupakan jenis *vending machine* yang sudah ada namun mengalami perubahan pada sistem pergerakan mekanik penggerak kaleng minuman yang lebih aman untuk menjaga minuman kaleng dan menggunakan *E-Money* sebagai alat bayar .

1.2. Identifikasi Masalah

Berdasarkan latar belakang di atas, masalah dapat diidentifikasi sebagai berikut:

- 1) Bagaimana merancang dan membangun *vending machine* menggunakan menggunakan Arduino Mega 2560 sebagai kontroler.
- 2) Bagaimana *actuator* wadah pengambil minuman (*box*) dapat bergerak menuju wadah penyimpanan minuman (*storage*) sesuai tombol pilihan minuman yang ditekan lalu mengambilnya untuk di sediakan kepada pembeli minuman kaleng.
- 3) Bagaimana merancang dan membuat sistem transaksi *E-Money* menggunakan kartu RFID , RFID reader, Database MySQL Workbench, serta aplikasi antar muka (*interface*) QT untuk administrator penjual saldo kartu *E-Money*.

- 4) Apakah isi saldo pada kartu *E-Money* bisa berkurang sejumlah harga barang setelah melakukan transaksi melalui *RFID reader* yang terpasang pada unit *vending machine*.
- 5) Bagaimana rancangan mekanik sistem agar dalam proses penyediaan produk tetap aman dan terhindar dari kerusakan.

1.3. Pembatasan Masalah

Dengan mempertimbangkan latar belakang dan Identifikasi masalah, maka pada penelitian prototipe *vending machine* berbasis Arduino Mega 2560 masalah dibatasi pada:

- 1) Perancangan dan membangun *vending machine* menggunakan menggunakan Arduino Mega 2560 sebagai kontroler.
- 2) Perancangan sistem kerja *actuator* wadah pengambil minuman (*box*) yang dapat bergerak menuju wadah penyimpanan minuman (*storage*) sesuai tombol pilihan minuman yang ditekan dan kemudian kembali ke lokasi awal untuk menyediakan minuman kepada pembeli minuman kaleng.
- 3) Perancangan dan pembuatan sistem emoney menggunakan kartu *RFID* sebagai alat bayar, *RFID reader* untuk mengidentifikasi kartu, database menggunakan MySQL Workbench, serta aplikasi antar muka (*interface*) QT untuk administrator penjual saldo kartu *E-Money*.
- 4) Perancangan protokoler *E-Money* agar saldo di database berkurang sejumlah harga barang setelah melakukan transaksi melalui *RFID reader* yang terpasang pada unit *vending machine*.
- 5) Perancangan dan pembuatan mekanik sistem agar dalam proses penyediaan produk tetap aman dan terhindar dari kerusakan.

1.4. Rumusan Masalah

Berdasarkan latar belakang masalah, identifikasi masalah, dan pembatasan masalah di atas maka masalah yang akan dibahas dalam penelitian ini dapat dirumuskan sebagai berikut: “Bagaimana merancang, membuat, dan menguji prototipe vending machine berbasis Arduino Mega 2560 dengan alat pembayaran E-Money?”

1.5. Tujuan Penelitian

Tujuan penelitian prototipe *vending machine* berbasis Arduino Mega 2560 antara lain:

- 1) Membuat prototipe *vending machine* untuk mengambil minuman kaleng otomatis menggunakan Arduino Mega 2560.
- 2) Membuat sistem *E-Money* menggunakan kartu RFID sebagai alat bayar transaksi *vending machine*.

1.6. Manfaat Penelitian

Hasil penelitian ini diharapkan akan berguna untuk beberapa hal, yakni:

1. Dengan adanya sistem kerja baru pada palat ini maka akan turut berkontribusi dalam hal menyempurnakan dan memperkaya varian baru dari *vending machine*.
2. Sebagai alat referensi untuk mempelajari sistem kerja *vending machine* menggunakan kendali Arduino Mega 2560 yang terdapat *E-Money* sebagai alat bayar.
3. Memberikan sumbangsih ide dan pemikiran yang dapat digunakan oleh mahasiswa khususnya mahasiswa Prodi.Pendidikan Teknik Elektronika

sebagai salah satu bahan kajian pengembangan dan pengaplikasian mikrokontroler Arduino Mega 2560.

4. Bagi peneliti, penelitian ini diharapkan mampu menambah wawasan dan pengetahuan yang berkenaan dengan proses pembuatan *vending machine*, aplikasi *interface* menggunakan software Qt creator, database menggunakan MySQL Workbench, dan sistem *E-Money*.

BAB II

TINJAUAN PUSTAKA

2.1. Prototipe

Menurut Kamus Besar Bahasa Indonesia prototype adalah model yang mula-mula (model asli) yang menjadi contoh (<http://kbbi.web.id/prototipe>). Definisi lain menjelaskan bahwa prototipe adalah Bentuk fisik pertama dari satu objek yang direncanakan dibuat dalam satu proses produksi, mewakili bentuk dan dimensi dari objek yang diwakilinya dan digunakan untuk objek penelitian dan pengembangan lebih lanjut.

Kriteria dari prototipe :

- 1) Bentuk awal dari objek yang akan diproduksi dalam jumlah banyak;
- 2) Prototipe dibuat berdasarkan pesanan untuk tujuan komersialisasi;
- 3) Belum pernah dibuat sebelumnya;
- 4) Merupakan hasil penelitian dan pengembangan dari objek atau sistem yang direncanakan akan dibuat;
- 5) Mudah dipahami dan dianalisis untuk pengembangan lebih lanjut.

(Sobri, <http://kip.bppt.go.id/index.php/prototipe/definisi-alih-teknologi>)

2.2. *Vending machine*

2.2.1. Konsep *Vending machine*

Mesin penjual otomatis/*Vending machine* (VM) didefinisikan sebagai mesin otomatis mandiri yang mengeluarkan barang (seperti permen dan minuman ringan) atau memberikan layanan (seperti timbangan timbangan) saat koin dimasukkan (Segrave, 2002: viii).

2.2.2. Sejarah *Vending Machine* (VM)

Referensi perangkat VM paling awal lahir dari Hero (Heron), seorang matematikawan, fisikawan, dan insinyur Yunani yang tinggal di Alexandria yang hidup di awal Abad Masehi. Dalam salah satu bukunya, *Pneumatica*, Hero menggambarkan sebuah alat yang dioperasikan dengan koin untuk digunakan menjual air pengorbanan di kuil Mesir. Selain kurangnya pengetahuan tentang siapa yang merancang mesin, juga tidak jelas apakah ada yang benar-benar diproduksi dan digunakan. Mungkin perangkat ini tidak pernah melampaui titik desain gambar (Segrave, 2002: 1).

Pada tahun 1830-an di Inggris versi yang lebih modern ditemukan, lalu di produksi dan dioperasikan tahun 1840-an di New York, dan didasarkan pada mesin Inggris sebelumnya. Paten pertama yang dikeluarkan untuk perangkat penjualan otomatis ternyata mengarah ke Simeon Denham dari Wakefield Yorkshire, England, tahun 1857. Pada tahun itu dia mengajukan permohonan paten untuk “mesin *Self-Acting* untuk pengiriman pos dan perangkong penerimaan,” yang dirancang untuk menjual stempel satu sen dari sebotol prangkong yang terdapat di dalamnya, lalu Denham diberi perlindungan sementara untuk penemuannya. Pada tahun-tahun berikutnya masih di abad yang sama terus dikeluarkan paten untuk menjual sapatangan, rokok, permen karet, dan lain-lain (Segrave, 2002: 4 - 5). Contoh bentuk mesin jual otomatis yang pernah diproduksi terlihat seperti yang ditunjukkan Gambar 2.1.



Gambar 2.1. mesin *self-acting* untuk pengiriman pos serta perangkat penerimaan di Inggris dan mesin jual permen karet berbentuk badut di Amerika

Di Jepang banyak *vending machine* yang tersedia dan beragam jenisnya. Bahkan di negara tersebut mudah sekali menjumpai *vending machine* di kota dan bahkan di desa. Negara ini termasuk yang memiliki kerapatan jumlah *vending machine* terbanyak di dunia (<http://www.japan-guide.com/e/e2010.html>). Gambar 2.2 memperlihatkan contoh *vending machine* yang ada di Jepang.



Gambar 2.2 *Vending machine* yang ada di Jepang

2.2.3. *Vending machine* di Indonesia

Di Indonesia, *vending machine* belum terlalu banyak jumlahnya, tetapi belakangan ini perkembangannya cukup pesat.

Dalam perkembangannya, *vending machine* sekarang mulai banyak ditemui terutama di kota-kota besar seperti Jakarta. Kebanyakan ditempatkan di fasilitas-fasilitas umum seperti bandara, stasiun, kompleks perbelanjaan besar maupun di kompleks sekolah.

Vending machine kebanyakan didatangkan atau diimport dari luar negeri, seperti Jepang, Korea, China, Amerika, atau Eropa. PT METEC Semarang adalah satu-satunya perusahaan pembuat *vending machine* di Indonesia (<http://kubota-ptms.co.id/indonesia/>).

2.3. Arduino

2.3.1. Definisi Arduino

Arduino adalah sebuah platform open source yang terdiri dari dua bagian utama yaitu perangkat keras Arduino (*hardware*) dan perangkat lunak (*software*). Papan Arduino (*Arduino hardware*) adalah mikrokontroler kecil yang di dalamnya terdapat chip kecil (*chip mikrokontroler*) yang merupakan jantung dari papan arduino. Logo Arduino dapat dilihat pada gambar 2.3.



Gambar 2.3 Logo Arduino

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

Pada awalnya Arduino merupakan proyek penelitian oleh Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino, dan David Mellis di awal tahun 2000 yang dibangun atas proyek Processing (sebuah bahasa pemrograman). Arduino bisa digunakan untuk mengembangkan objek interaktif mandiri yang

dapat dihubungkan ke computer. *Sketch* program yang akan dirancang dapat dibuat menggunakan Arduino IDE (*Integrated Development Environment*) untuk selanjutnya *sketch* dapat dimasukkan ke dalam papan Arduino. Arduino IDE dapat diunduh secara gratis melalui website resmi www.arduino.cc di internet (Banzi, 2011).

2.3.2. Arduino Mega 2560

Arduino MEGA 2560 adalah papan mikrokontroler berdasarkan ATmega2560 yang memiliki 54 pin input/output digital (15 dapat digunakan sebagai output PWM), 16 input analog, 4 UART (port serial perangkat keras), osilator kristal 16 MHz, koneksi USB, colokan listrik, header ICSP, Dan tombol reset. Dengan kapasitas yang lebih besar maka papan Arduino Mega 2560 direkomendasikan untuk printer 3D dan proyek robotika. Papan Arduino Mega 2560 merupakan edisi perbaikan yang menggantikan papan Arduino Mega yang masih menggunakan Atmega (<https://store.arduino.cc/usa/arduino-mega-2560-rev3>). *Board* arduino ditunjukkan pada gambar 2.4, dan spesifikasinya ditampilkan pada table 2.1.



Gambar 2.4 Papan Arduino Mega 2560

(sumber: <https://store.arduino.cc/usa/arduino-mega-2560-rev3>)

Tabel 2.1 Spesifikasi Arduino Mega 2560

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7 – 12V
Input Voltage (limits)	6 – 20V
Digital I/O Pins	54 (of which 14 provide PWM output)
Analog Input Pins	16
DC Current per I/O pin	40 mA
DC Current for 3.3V pin	50 mA
Flash Memory	256 KB of which 8 KB used bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

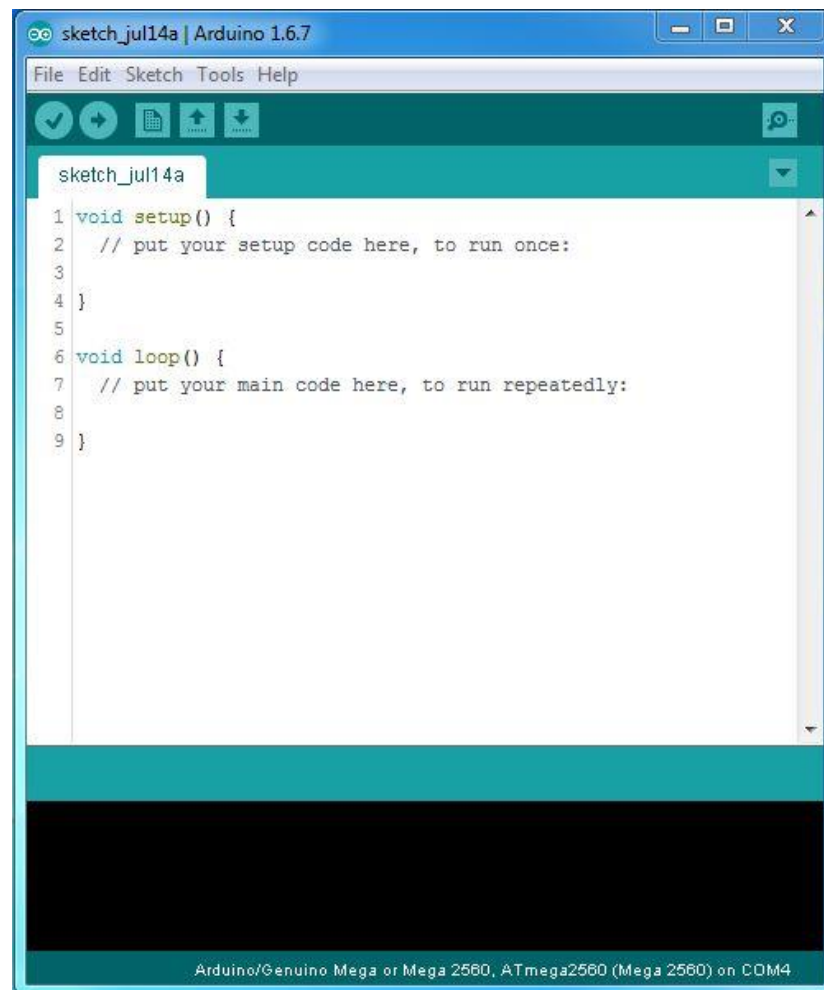
(sumber: <http://www.robotshop.com/media/files/pdf/arduinomega2560datasheet.pdf>)

2.3.3. Arduino Software IDE

IDE (Integrated Development Environment) adalah program khusus yang berjalan (*running*) pada *Personal Computer (PC)* yang memungkinkan untuk menulis *Sketch* (sebutan untuk lembar kerja program pada arduino) untuk di-*upload* ke papan Arduino dengan model bahasa sederhana yang diolah dengan bahasa *Processing* (Banzi, 2011: 20).

Arduino IDE berisi editor teks untuk menulis kode, area pesan, konsol teks, toolbar dengan tombol untuk fungsi umum dan serangkaian menu. Ini terhubung ke perangkat keras Arduino untuk mengunggah program dan berkomunikasi dengannya. Program yang ditulis menggunakan Arduino Software (IDE) disebut sketsa. Sketsa ini ditulis dalam editor teks dan disimpan dengan ekstensi file *.ino*. Editor memiliki fitur untuk memotong / menempel dan mencari / mengganti teks.

Area pesan memberi umpan balik sambil menyimpan dan mengekspor dan juga menampilkan kesalahan. Konsol menampilkan output teks oleh Arduino Software (IDE), termasuk pesan kesalahan lengkap dan informasi lainnya (<https://www.arduino.cc/en/Guide/Environment>). Tampilan Software Arduino IDE terlihat seperti gambar 2.5.



Gambar 2.5 Screenshot Tampilan Arduino IDE pada PC

(Sumber: *screenshot/dokumentasi pribadi*)

2.4. E-Money

E-Money adalah uang elektronik yang ditukar secara elektronik melalui perangkat teknis seperti komputer atau telepon genggam. *E-Money* yang beredar beroperasi sebagai instrumen pembawa pra-bayar. Contoh *E-Money* paling

terkenal adalah Bitcoin, yang bisa dibeli dengan uang sungguhan dan diperdagangkan di bursa seperti mata uang lainnya (http://lexicon.ft.com/Term?term=e_money)

Sebagai istilah "uang elektronik" sudah menyiratkan, *E-Money* adalah uang yang tersimpan secara elektronik pada perangkat atau dari jarak jauh di server. Jenis *E-Money* yang umum mencakup bentuk akun online dan voucher Prabayar tertentu seperti Paysafecard (<https://www.paysafecard.com/en/corporate/blog/detail/what-is-E-Money/>).

2.5. MySQL Workbench

MySQL Workbench adalah alat visual terpadu untuk arsitek, pengembang, dan database administrators (DBA) database. MySQL Workbench menyediakan pemodelan data, pengembangan SQL, dan alat administrasi yang komprehensif untuk konfigurasi server, administrasi pengguna, backup, dan masih banyak lagi. MySQL Workbench tersedia di Windows, Linux dan Mac OS X. Logo Aplikasi MySQL Workbench dapat dilihat pada gambar 2.6.



Gambar 2.6 Logo MySQL Workbench

MySQL Workbench memungkinkan seorang DBA, pengembang, atau arsitek data untuk merancang, membuat, dan mengelola database secara visual. Ini mencakup semua kebutuhan pemodelan data untuk menciptakan model ER yang

kompleks, maju dan rekayasa mundur, dan juga memberikan fitur utama untuk melakukan tugas manajemen dan dokumentasi perubahan yang sulit yang biasanya memerlukan banyak waktu dan usaha.

MySQL Workbench memberikan alat visual untuk membuat, mengeksekusi, dan mengoptimalkan query SQL. Editor SQL memberikan penyorotan sintaks warna, melengkapi secara otomatis, menggunakan kembali cuplikan SQL, dan riwayat eksekusi SQL. Database Connections Panel memungkinkan pengembang untuk dengan mudah mengelola koneksi database standar, termasuk MySQL Fabric. Object Browser menyediakan akses cepat ke skema database dan objek.

MySQL Workbench menyediakan konsol visual untuk mengelola lingkungan MySQL dengan mudah dan mendapatkan visibilitas yang lebih baik ke dalam database. Pengembang dan DBA dapat menggunakan alat visual untuk mengonfigurasi server, mengelola pengguna, melakukan backup dan recovery, memeriksa data audit, dan melihat kesehatan database (<https://www.mysql.com/products/workbench/>).

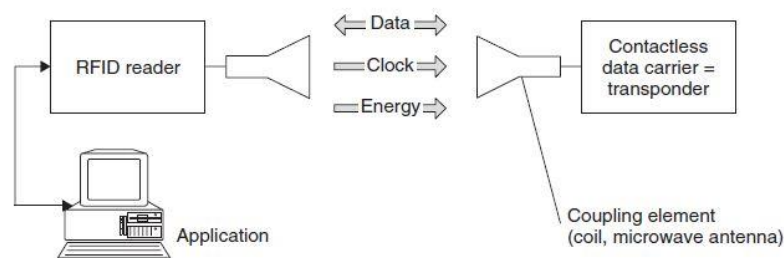
2.6. RFID

Radio frequency identification (RFID) adalah sebuah teknologi yang menggunakan komunikasi via gelombang elektromagnetik untuk merubah data antara terminal dengan suatu objek seperti produk barang, hewan, ataupun manusia dengan tujuan untuk identifikasi dan penelusuran jejak melalui penggunaan suatu piranti yang bernama RFID tag. RFID yang digunakan pada penelitian ini adalah RFID *reader* MFRC522 dan kartu RFID MIFARE yang

beroperasi pada frekuensi 13,56 Hz. Data yang terbaca oleh RFID *reader* MFRC522 yaitu *hexadecimal*.

RFID tag dapat bersifat aktif atau pasif. RFID tag yang pasif tidak memiliki power supply sendiri, sehingga harganya pun lebih murah dibandingkan dengan tag yang aktif. Dengan hanya berbekal induksi listrik yang ada pada antenna yang disebabkan oleh adanya pemindaian frekuensi radio yang masuk, sudah cukup untuk memberi kekuatan yang cukup bagi RFID tag untuk mengirimkan respon balik. Dengan tidak adanya power supply pada RFID tag yang pasif maka akan menyebabkan semakin kecilnya ukuran dari RFID tag yang mungkin dibuat, bahkan lebih tipis daripada selembar kertas dengan jarak jangkauan yang berbeda mulai dari 10 mm sampai dengan 6 meter. RFID tag yang aktif memiliki power supply sendiri dan memiliki jarak jangkauan yang lebih jauh. Memori yang dimilikinya juga lebih besar sehingga bisa menampung berbagai macam informasi di dalamnya. RFID tag yang banyak beredar sekarang adalah RFID tag yang sifatnya pasif.

Sistem RFID selalu terdiri dari dua komponen, dan interaksi antara kedua komponen terlihat seperti ilustrasi pada gambar 2.7.



Gambar 2.7 Ilustrasi Interaksi Komponen RFID

(Sumber: RFID Handbook Fundamentals and Applications in Contactless Smart Cards, Radio Frequency Identification and Near-Field Communication, Third Edition)

Penggunaan Contactless data carrier (kartu RFID) sangat mudah hanya dengan menempelkan/ mendekatkan dengan jarak tertentu kartu tersebut ke RFID *reader* maka sudah terjalin komunikasi data. Gambar 2.8 memperlihatkan contoh ilustrasi penggunaan kartu RFID yang di tempelkan ke RFID *reader*.



Gambar 2.8 Penggunaan RFID reader dan contactless smart card digunakan secara praktis

(Sumber: RFID Handbook Fundamentals and Applications in Contactless Smart Cards, Radio Frequency Identification and Near-Field Communication, Third Edition)

Kegunaan dari sistem RFID ini adalah untuk mengirimkan data dari tag yang kemudian dibaca oleh RFID reader dan kemudian diproses oleh aplikasi computer. Data yang dipancarkan dan dikirimkan tadi bisa berisi beragam informasi, seperti ID, informasi lokasi atau informasi lainnya.

Dalam suatu sistem RFID sederhana, suatu objek dilengkapi dengan tag yang berisi *microchip* yang ditanamkan di dalamnya yang berisi sebuah kode produk yang sifatnya unik. Sebaliknya, interrogator, suatu antena yang berisi transceiver dan decoder, memancarkan sinyal yang bisa mengaktifkan RFID tag sehingga dapat membaca dan menulis data ke dalamnya. Ketika suatu RFID tag melewati suatu zone elektromagnetis, maka akan mendeteksi sinyal

aktivasi yang dipancarkan oleh reader. Reader akan melakukan *decode* data yang ada pada tag dan kemudian data tadi akan diproses oleh komputer.

RFID tag seringkali dianggap sebagai pengganti dari barcode. RFID juga sudah diajukan untuk penggunaan pada point-of-sale yang menggantikan kasir dengan suatu mesin otomatis tanpa harus melakukan barcode scanning (Yulius, <http://sis.binus.ac.id/2014/04/12/radio-frequency-identification-rfid/>).

2.7. Text LCD 16x2

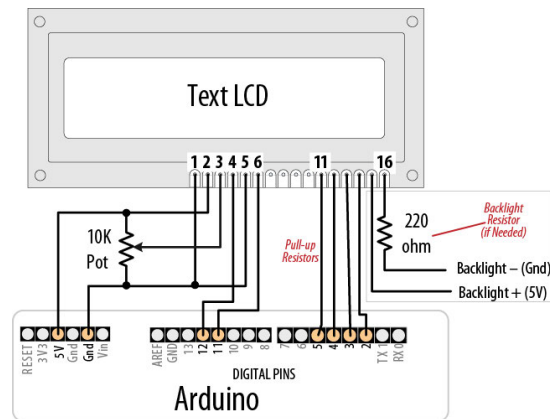
LCD 16x2 yang paling populer adalah panel teks berdasarkan chip Hitachi HD44780 Driver. LCD ini menampilkan dua atau empat baris teks, dengan 16, 20, 32, atau 40 karakter. Pada Arduino terdapat sebuah *Library* untuk mengemudikan (*driving*) display LCD teks Arduino, dan dapat mencetak teks di LCD (Margolis, 2011: 333).

Text LCD yang beredar dipasaran dan populer salah satu contohnya adalah Text LCD 16x2 seperti terlihat pada gambar 2.9, lalu gambar 2.10 menampilkan contoh instalasi Text LCD pada Arduino.



Gambar 2.9 Text LCD 16x2

(sumber: <https://www.arduino.cc/en/Tutorial/HelloWorld>)



Gambar 2.10 Contoh Instalasi Text LCD pada Papan Arduino

(sumber: *Arduino Cookbook*)

2.8. I2C (*Inter Integrated Circuit*)

I2C (*Inter Integrated Circuit*) adalah sebuah protokol untuk komunikasi serial antar IC, dan sering disebut juga Two Wire Interface (TWI). Bus yang digunakan untuk komunikasi antara mikrokontroler dan *peripheral device* seperti memori, sensor temperatur dan I/O expander (Tjahyadi, <http://christianto.tjahyadi.com/belajar-mikrokontroler/komunikasi-i2c.html>).

Standar I2C (Inter-Integrated Circuit) dan SPI (Serial Peripheral Interface) diciptakan untuk menyediakan cara sederhana agar informasi digital dapat ditransfer antara sensor dan mikrokontroler seperti Arduino. Library Arduino untuk kedua I2C dan SPI memudahkan untuk menggunakan kedua protokol ini. Pilihan antara I2C dan SPI biasanya ditentukan oleh perangkat yang ingin Anda hubungkan. Beberapa perangkat menyediakan kedua standar, namun biasanya perangkat atau chip mendukung satu atau yang lain.

Untuk penggunaan IC2 pada LCD terdapat modul khusus LCD yang disebut SPI character LCD backpack. LCD Backpack mengurangi jumlah pin yang dibutuhkan untuk terhubung ke LCD. Dengan menggunakan i2c dan SPI

input / output expanders yang sederhana bisa mengurangi jumlah pin (hanya 2 pin yang dibutuhkan untuk i2c) namun tetap mudah untuk berinteraksi dengan LCD. Bagi pengguna Arduino, telah tersedia Library yang mudah digunakan dan kompatibel dengan proyek menggunakan kabel '6 pin'. Backpack ini akan bekerja dengan LCD karakter 'standar', dari ukuran 8x1 sampai 20x4 (<https://www.adafruit.com/product/292>). Modul *Backpack* I2C yang sudah terpasang pada Text LCD terlihat seperti gambar 2.11.



Gambar 2.11 I2C LCD Backpack yang sudah terpasang pada Text LCD

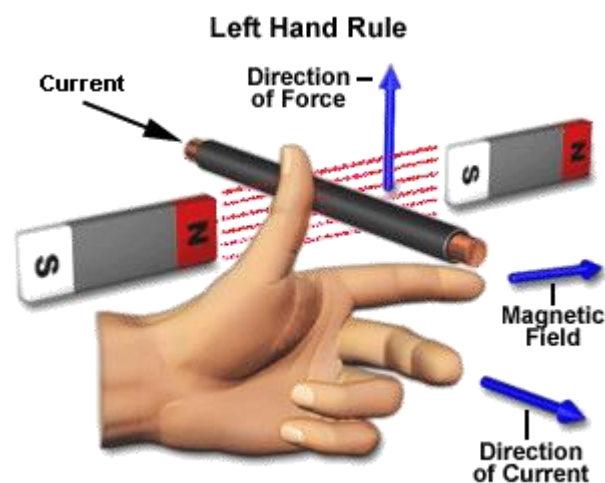
(Sumber: <http://www.belajarduino.com/2016/06/how-to-connect-1602-2004-iic-i2c-lcd-to.html>)

2.9. Motor DC

Motor Direct Current (DC) merupakan jenis motor listrik yang mengkonversi energy listrik DC menjadi energy mekanik. Prinsip Motor DC atau motor arus searah ini bekerja dengan prinsip ketika konduktor pembawa arus ditempatkan di medan magnet maka akan mengalami torsi dan memiliki kecenderungan untuk bergerak. Ini dikenal dengan aksi otomotif. Jika arah arus di kawat dibalik, arah rotasi juga membalikkan. Ketika medan magnet dan medan

listrik berinteraksi, mereka menghasilkan gaya mekanik, dan berdasarkan prinsip kerja motor DC (<https://www.electrical4u.com/dc-motor-or-direct-current-motor/>).

Arah rotasi motor ini diberikan oleh peraturan tangan kiri Fleming, yang menyatakan bahwa jika jari telunjuk, jari tengah dan ibu jari tangan kiri Anda saling bertautan saling tegak lurus satu sama lain dan jika jari telunjuknya mewakili arah medan magnet , Jari tengah menunjukkan arah arus, maka jempol mewakili arah di mana gaya dialami oleh poros motor DC seperti ilustrasi pada gambar 2.12.



Gambar 2.12 Aturan Tangan Kiri Fleming untuk Hubungan Arah Arus ,Medan Magnet, dan Gaya

(Sumber: <https://www.electrical4u.com/dc-motor-or-direct-current-motor/>)

Gaya pada kawat dengan panjang l , membawa arus I dan terpapar medan magnetik B diberikan oleh ungkapan sederhana:

$$\mathbf{F} = \mathbf{B} \mathbf{I} l$$

Dimana F berada di newton saat B berada dalam tesla, I di ampere, dan l dalam meter.

Ketatnya persamaan Gaya (F) diatas hanya berlaku bila arus tegak lurus terhadap medan. Jika kondisi ini tidak terpenuhi, kekuatan pada konduktor akan berkurang; Dan dalam kasus ekstrim di mana arus berada dalam arah yang sama dengan medan, kekuatannya akan turun menjadi nol. Setiap perancang motor tahu bahwa untuk mendapatkan yang terbaik dari medan magnetik, harus tegak lurus terhadap konduktor, dan karena itu sebaiknya diasumsikan selanjutnya bahwa B dan I selalu tegak lurus (Hughes, 2006: 6-7). Contoh fisik gambar motor DC ditunjukkan pada gambar 2.13.



Gambar 2.13 Motor DC

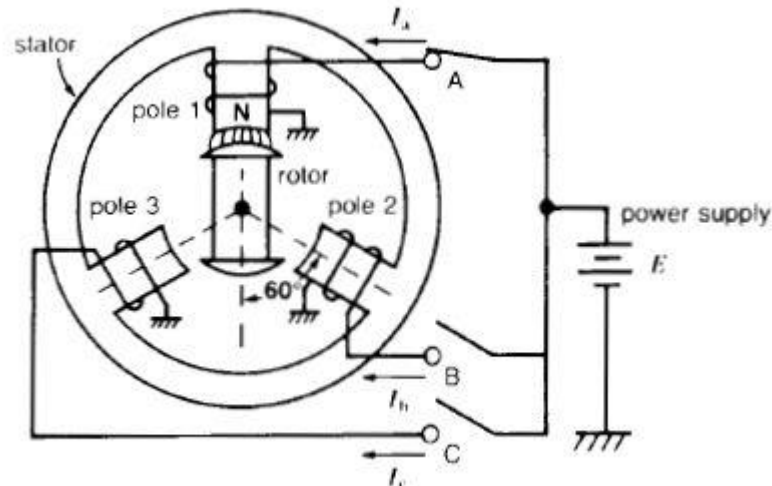
(Sumber: <http://www.nextpowerelectronics.com/dc-motor.htm>)

2.10. Motor Stepper

Motor stepper adalah motor khusus yang digunakan saat gerak dan posisi harus dikontrol dengan tepat. Seperti namanya, motor stepper berputar dalam langkah-langkah terpisah, setiap langkah sesuai dengan pulsa yang dipasok ke salah satu gulungan statornya. Bergantung pada desainnya, motor stepper bisa melaju 90° , 45° , 16° , atau sedikit pecahan per detik. Dengan memvariasikan denyut nadi, motor dapat dibuat maju dengan sangat lambat, satu langkah pada satu waktu, atau berputar secara bertahap pada kecepatan setinggi 4000 r/menit.

Motor stepper bisa berputar searah jarum jam atau berlawanan arah jarum jam, tergantung urutan pulsa yang diaplikasikan pada gulungan (Wildi, 2002: 417).

Contoh motor stepper sederhana dapat dilihat pada gambar 2.14.



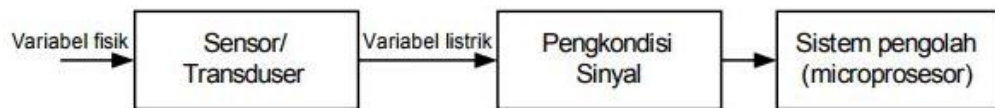
Gambar 2.14 Motor Stepper Sederhana Dimana Setiap Langkah Menggerakkan Rotor Sebesar 60 derajat

(Sumber: *Electrical Machines, Drives, and Power Systems, Fifth Edition*)

2.11. Sensor

2.11.1. Definisi Sensor

Dalam kaitannya dengan sistem elektronis, Sensor dan transduser pada dasarnya dapat dipandang sebagai sebuah perangkat (*device*) yang berfungsi mengubah suatu besaran fisik menjadi besaran listrik, sehingga keluarannya dapat diolah dengan rangkaian listrik atau sistem digital (Setiawan, 2009: 1). Gambar 2.15 menjelaskan tentang diagram blok fungsional sensor/transduser.



Gambar 2.15 Blok Fungsional Sensor/Transduser

(Sumber: http://eprints.undip.ac.id/4886/1/Sensor_dan_Transduser.pdf)

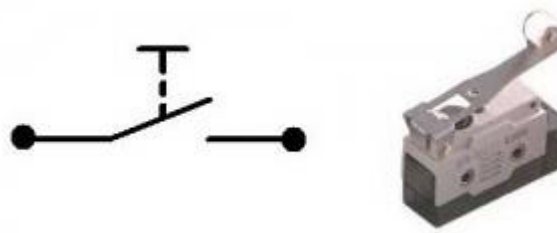
2.11.2. Jenis-jenis Sensor

Berdasarkan variabel yang diindranya, sensor dikategorikan kedalam dua jenis, yaitu sensor fisika dan sensor kimia. Sensor fisika merupakan jenis sensor yang mendeteksi suatu besaran berdasarkan hukum-hukum fisika, yaitu seperti sensor cahaya, suara, gaya, kecepatan, percepatan, maupun sensor suhu.

Sensor digunakan dalam kehidupan sehari-hari, dimana aplikasinya mencakup berbagai bidang, yaitu seperti: automobile, mesin, kedokteran, industri, robot, maupun aerospace. Dalam lingkungan sistem kontrol dan robotika, sensor memberi fungsi seperti layaknya mata, pendengaran, hidung, maupun lidah yang kemudian akan diolah oleh kontroller sebagai otaknya (Setiawan, 2009: 1-3).

2.11.3. Sensor Mekanik

Sensor Mekanik (*Mechanics Sensor*) merupakan sensor atau transduser yang digunakan untuk mengetahui, mengukur atau mendeteksi nilai perubahan atau gerakan mekanis dari suatu objek (<http://elektronika-dasar.web.id/pengertian-dan-jenis-sensor-mekanik-mechanics-sensor/>). Sensor mekanik dapat dilihat pada gambar 2.16.

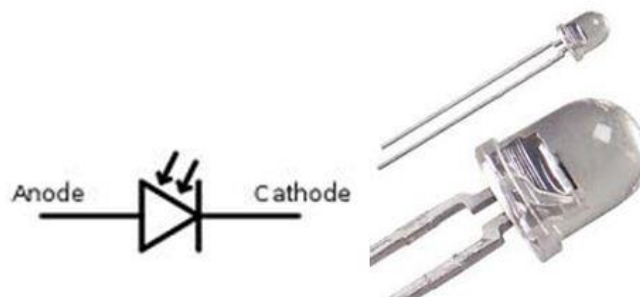


Gambar 2.16 Limit Switch, Sensor Mekanik Pendeteksi Batas Gerakan Suatu Objek

(Sumber: <http://belajarelektronika.net/perbedaan-limit-switch-dan-saklar-push-on/>)

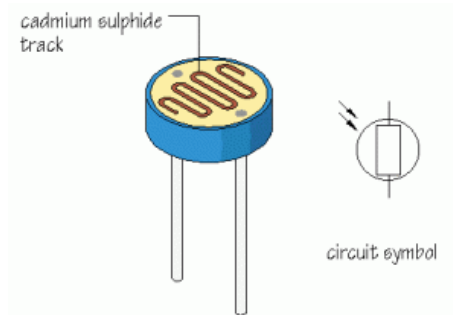
2.11.4. Sensor Cahaya

Sensor cahaya adalah sensor yang dapat melakukan pendeteksian cahaya, lalu melakukan perubahan terhadapnya jadi sinyal listrik dan dipakai dalam sebuah rangkaian yang memakai cahaya sebagai pemicunya. Beberapa komponen yang biasanya digunakan dalam rangkaian sensor cahaya diantaranya Light Dependent Resistor / LDR, Photodiode/ dioda foto, dan Photo Transistor / Foto Transistor (<http://elektronikadasar.info/sensor-cahaya.htm>). Simbol dan fisik fotodioda dapat dilihat pada gambar 2.17, simbol dan fisik LDR dapat dilihat pada gambar 2.18, simbol dan fisik fototransistor dapat dilihat pada gambar 2.19.



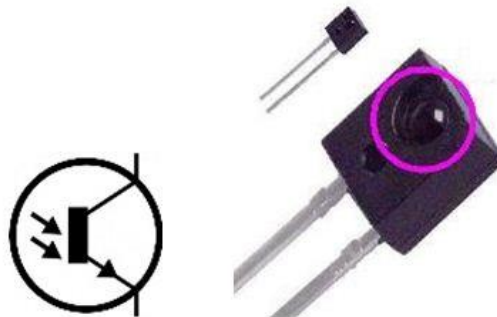
Gambar 2.17 Simbol dan Bentuk Fisik Fotodioda

(Sumber: <http://www.edukasiElektronika.com/2013/02/photo-dioda.html>)



Gambar 2.18 Simbol dan Bentuk Fisik Fotodiode

(Sumber: <http://elektronikadasar.info/sensor-cahaya.htm>)



Gambar 2.19 Simbol dan Bentuk Fisik Fototransistor

(Sumber: <http://elektronikadasar.info/sensor-cahaya.htm>)

2.12. A4988 Driver Motor Stepper

A4988 adalah driver microstepping untuk mengendalikan motor stepper bipolar yang memiliki penerjemah built-in untuk pengoperasian yang mudah. Ini berarti kita bisa mengendalikan motor stepper dengan hanya 2 pin dari controller kita, atau satu untuk mengendalikan arah rotasi dan yang lainnya untuk mengendalikan langkah-langkahnya.

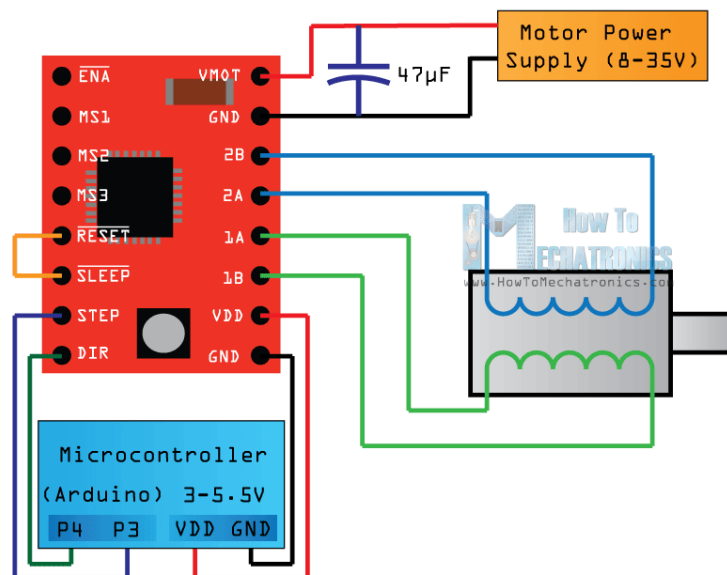
Driver memberikan lima resolusi langkah yang berbeda: *full-step*, *half-step*, *quarter-step*, *eight-step* dan *sixteenth-step*. Selain itu, ia memiliki potensiometer untuk menyesuaikan keluaran arus, penghentian termal suhu lebih tinggi dan

perlindungan arus silang. Tegangan logiknya adalah dari 3 sampai 5,5 V dan arus maksimum per fase adalah 2A jika penambahan pendinginan yang baik disediakan atau arus kontinyu 1A per fase tanpa heat sink atau pendinginan. Tabel 2.2 menjelaskan spesifikasi A4988.

Tabel 2.2 Spesifikasi A4988

Minimum Logic Voltage	3V
Maximum Logic Voltage	5.5 V
Continuous current per phase	1 A
Maximum current per phase	2 A
Minimum Operating Voltage	8 V
Maximum Operating Voltage	35 V

Pinout modul A4988 dijelaskan melalui gambar berikut:



Gambar 2.20 Pinout A4988 dan Wiring Diagram Minimum Sistem Driver

(Sumber: <http://howtomechatronics.com/tutorials/arduino/how-to-control-stepper-motor-with-a4988-driver-and-arduino/>)

Tiga pin berikutnya (MS1, MS2 dan MS3) dari gambar diatas adalah untuk memilih salah satu dari lima resolusi langkah sesuai tabel kebenaran di atas. Pin ini memiliki resistor pull-down internal jadi jika kita membiarkannya terputus, board akan beroperasi dalam mode full step. Tabel kebenaran pada driver A4988 dapat dilihat pada table 2.3.

Tabel 2.3 Pilihan Step Mode pada driver A4988

MS1	MS2	MS3	Resolution
LOW	LOW	LOW	Full Step
HIGH	LOW	LOW	Half Step
LOW	HIGH	LOW	Quarter Step
HIGH	HIGH	LOW	Eighth Step
HIGH	HIGH	HIGH	Sixteenth Step

Yang terakhir, pin ENABLE digunakan untuk menyalakan atau mematikan output FET. Jadi logika HIGH maka akan menjaga output dinonaktifkan.

BAB III

METODOLOGI PENELITIAN

3.1. Tempat dan Waktu Penelitian

Penelitian dilakukan di Gedung L, Teknik Elektro, Fakultas Teknik, Universitas Negeri Jakarta.

3.2. Alat dan Bahan Penelitian

3.2.1. Alat

Pembuatan prototipe *vending machine* ini menggunakan alat dan bahan sebagai berikut:

1. Alat Ukur:

- a. AVO meter/ Multitester
- b. Timbangan, untuk mengukur massa minuman kaleng (benda yang digerakkan *actuator*)
- c. Penggaris/ meteran, untuk mengukur jangkauan *actuator* sumbu X, Y

2. *Hardware*

- a. HP G42 Notebook PC
- b. Board Arduino Mega 2560
- c. Modul Driver Stepper dan Motor Stepper Nema 17
- d. Driver L293D dan motor DC *geared*
- e. Modul Sensor Cahaya

3. *Software*

- a. Arduino Software IDE
- b. MySQL 6.3 CE

- c. QT Creator 3.0.1
- d. Sketcup 2016
- e. Fritzing

4. Mekanik

Sebagian besar kerangka utama alat ini menggunakan :

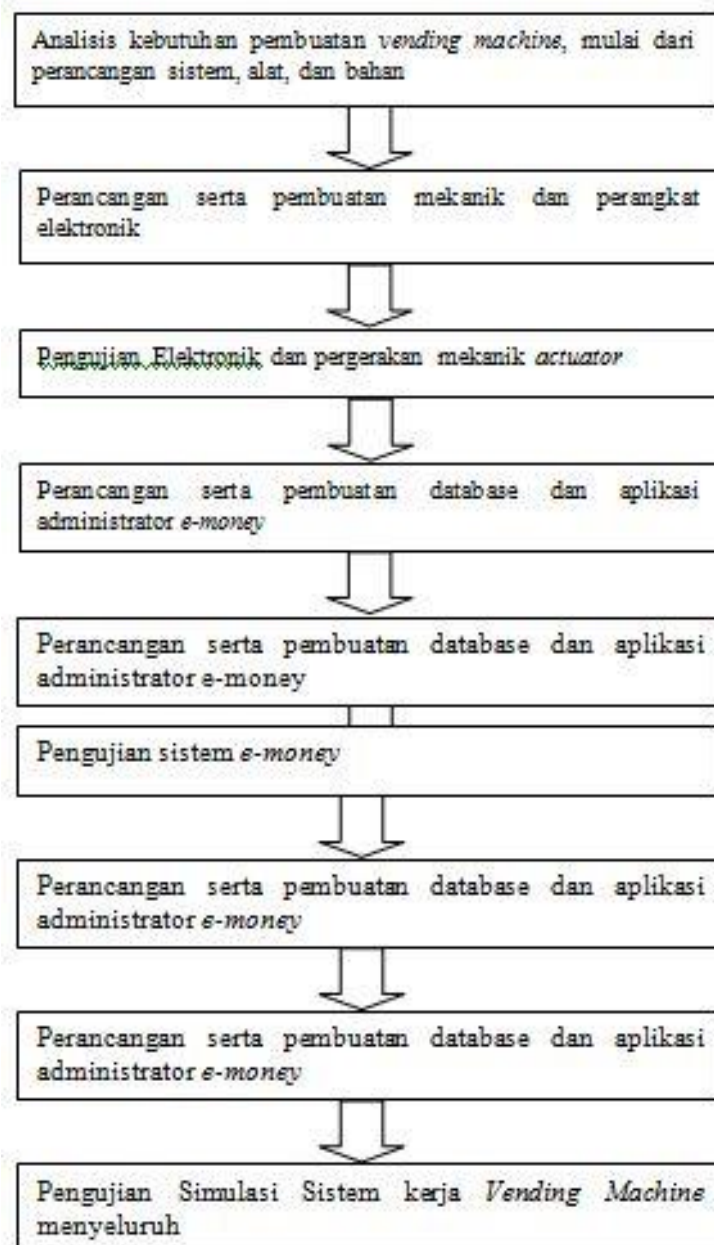
- a. Alumunium Profile 2020
- b. Alumunium Profile 2040
- c. V-Wheels Plate Kit
- d. Acrylic

3.2.2. Bahan

Bahan yang digunakan dalam penelitian ini adalah objek benda (minuman kaleng), data hasil transaksi dari database, dan hasil pengukuran pergerakan *actuator*.

3.3. Diagram Alir Penelitian

Penelitian ini membutuhkan tahapan-tahapan yang harus dilakukan sesuai prosedur untuk mendapatkan hasil yang diharapkan sesuai tujuan penelitian. Tahapan-tahapan tersebut sesuai dengan gambar 3.1.



Gambar 3.1. Diagram Alir Penelitian

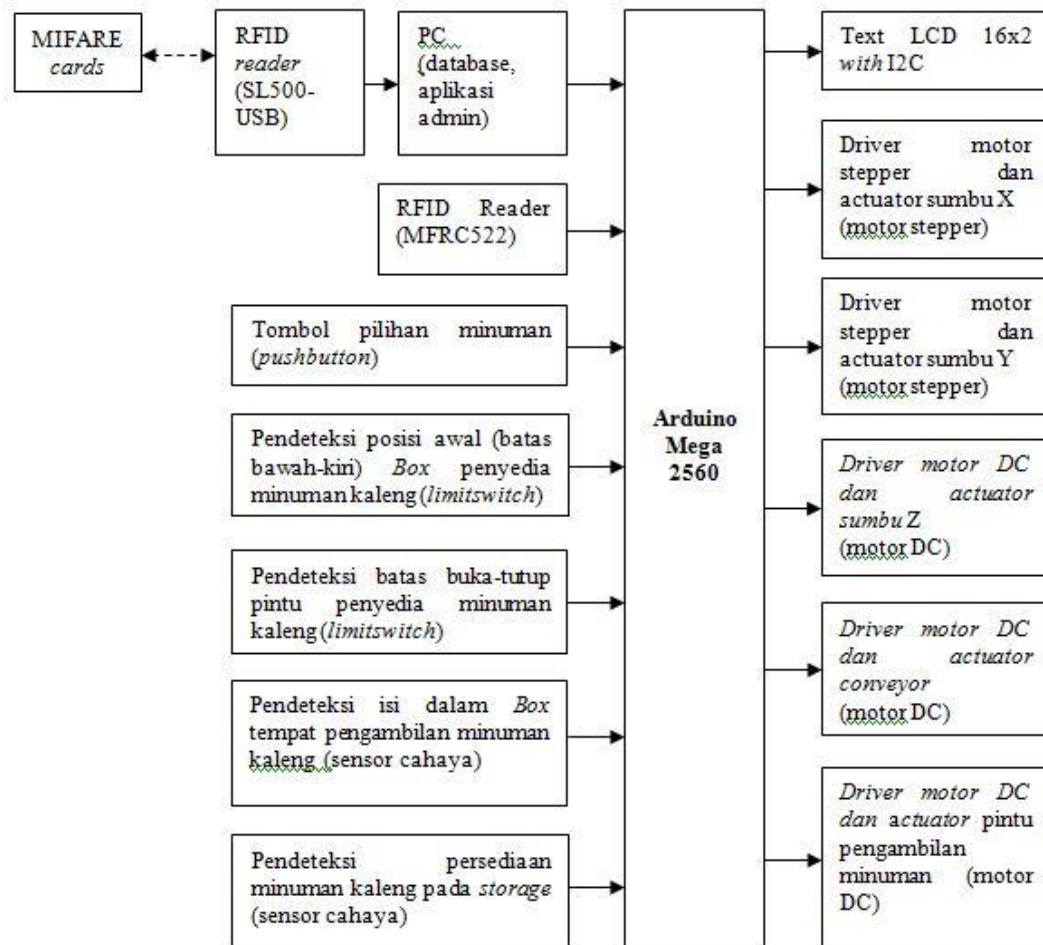
3.3.1. Tahap Analisis

Analisa sistem bertujuan untuk mengidentifikasi permasalahan-permasalahan yang ada pada sistem. Pada pembuatan *vending machine minuman kaleng* telah dianalisa kebutuhan-kebutuhan yang harus ada, yakni:

- 1) Komponen *hardware* elektronik yang meliputi: Mikrokontroler, motor stepper, driver motor stepper, motor DC geared/gearbox, driver motor DC
- 2) Komponen Software yang meliputi: Database, Aplikasi Administrator *E-Money*, firmware kontroler.
- 3) Komponen mekanik yang meliputi: mekanik *actuator* sumbu X, *actuator* sumbu Y, *actuator* sumbu Z, *actuator conveyor*, *conveyor* wadah penyimpanan (*storage*) minuman kaleng, wadah pengambilan (*box*) minuman kaleng, pintu tempat pengambilan minuman

3.3.2. Tahap Perancangan

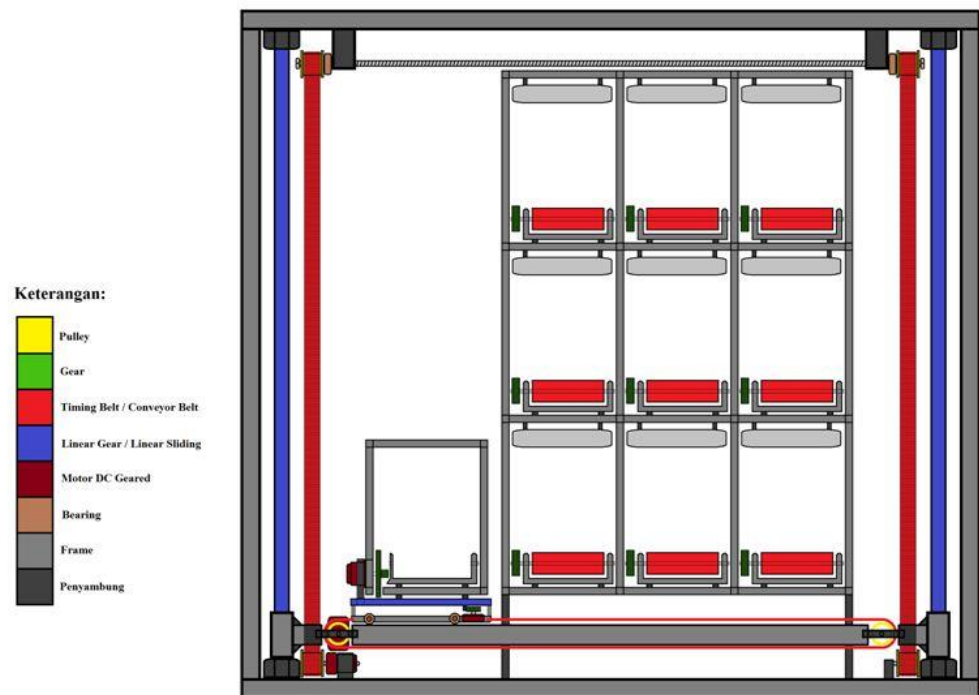
Perancangan dilakukan dalam rangka persiapan pembuatan alat. Rancangan dibuat agar sesuai dengan tujuan pembuatannya. Hubungan antar komponen pada sistem *vending machine* minuman kaleng dijelaskan melalui diagram blok pada gambar 3.2.



Gambar 3.2. Diagram Blok Sistem *Vending machine* Minuman Kaleng Berbasis Arduino Mega 2560

3.4. Rancangan Mekanik

Struktur mekanik utama yang dibutuhkan dalam rancangan mekanik terdiri dari *actuator* sumbu X, *actuator* sumbu Y, *actuator* sumbu Z, *actuator*, *Conveyor*, Pintu geser tempat pengambilan minuman kaleng, *Box* tempat pengambilan minuman kaleng, dan *Storage* tempat penyimpanan minuman kaleng (berupa *conveyor*). Rancangan mekanik *vending machine* dapat dilihat pada gambar 3.3.



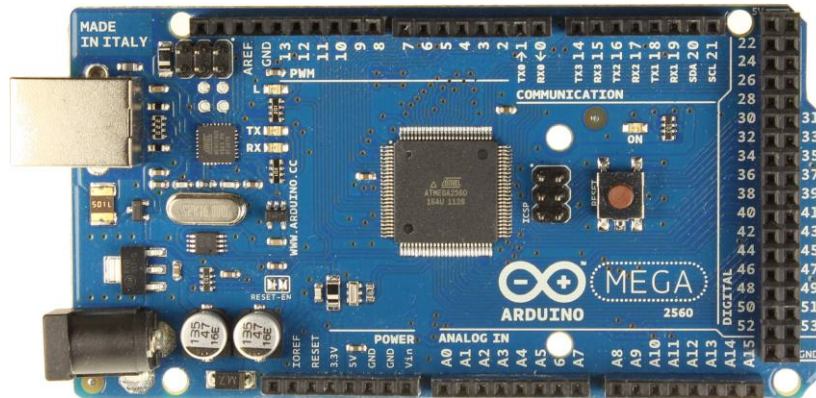
Gambar 3.3 Rancangan mekanik *vending machine*

3.5. Rancangan Elektronik

Dalam pembuatan alat prototipe *vending machine* ini diperlukan beberapa peralatan elektronik, maka dari itu diperlukan rancangan elektronik pendukung sistem ini. Hardware yang digunakan ada yang dirancang sendiri dan adapula yang menggunakan modul yang sudah ada. Perangkat elektronik yang dibutuhkan adalah sebagai berikut:

1) Board Arduino Mega 2560

Arduino berperan sebagai prosessor dalam sistem *vending machine* pada penelitian ini. Arduino akan dihubungkan dengan semua perangkat elektronik lainnya dalam sistem. Board Arduino Mega 2560 dapat dilihat pada gambar 3.4.



Gambar 3.4 Board arduino

2) Driver Motor Stepper A4988 dan Stepper Motor Nema 17

Motor Stepper digunakan untuk *actuator* sumbu X dan sumbu Y. Motor ini menyatu dengan mekanik pergerakan linear arah sumbu X dan sumbu Y. Motor stepper yang digunakan adalah motor stepper Nema 17 yang dapat dilihat pada gambar 3.5, lalu driver motor stepper menggunakan modul A4988 seperti yang terlihat pada gambar 3.6.

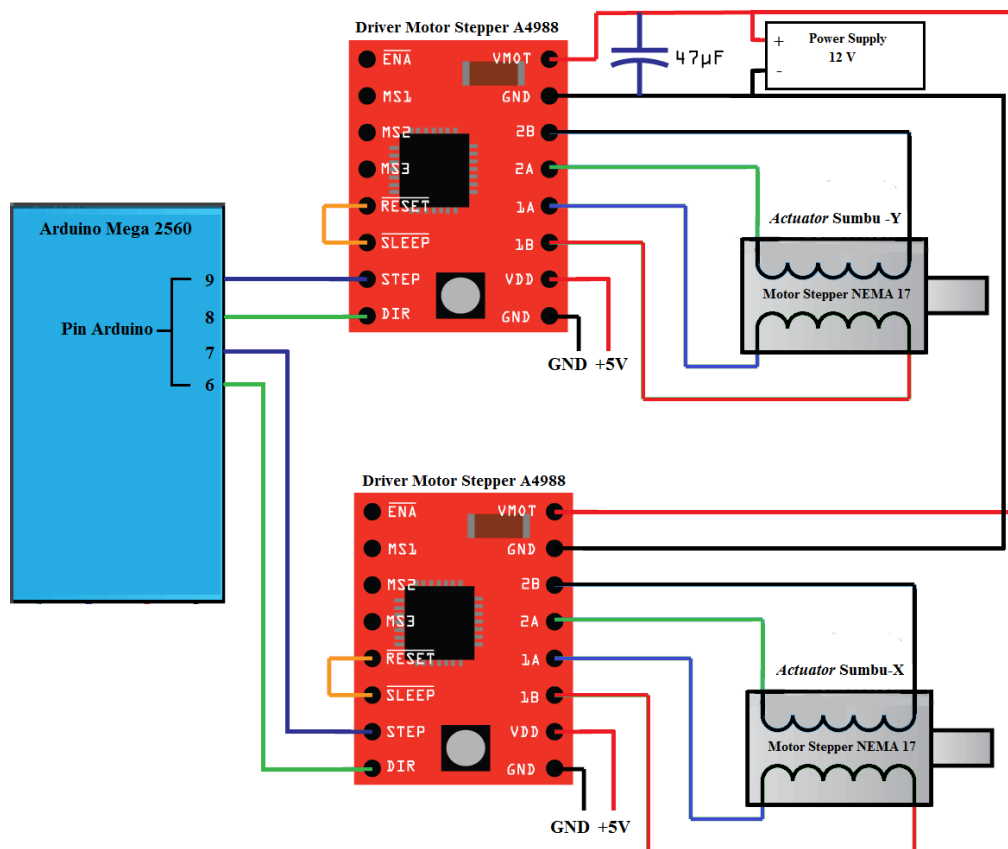


Gambar 3.5 Motor Stepper Nema 17



Gambar 3.6 Modul A4988

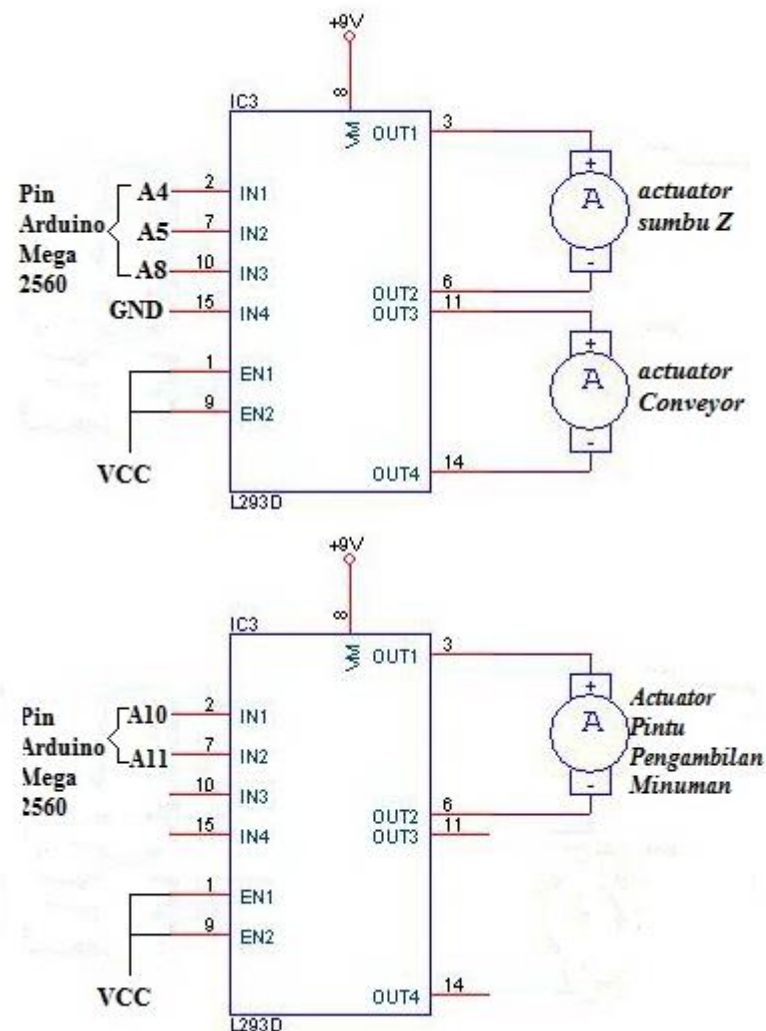
Motor Stepper dihubungkan ke driver motor stepper, selanjutnya motor stepper akan dikendalikan pergerakannya oleh sinyal dari Arduino Mega 2560. Gambar 3.7 memperlihatkan rancangan skematik rangkaian pengendali motor stepper sebagai *actuator* sumbu-X dan sumbu Y dengan pengendali Arduino Mega 2560.



Gambar 3.7 Skematik Pengendali Motor Stepper

3) Driver motor DC IC L293D dan motor DC geared

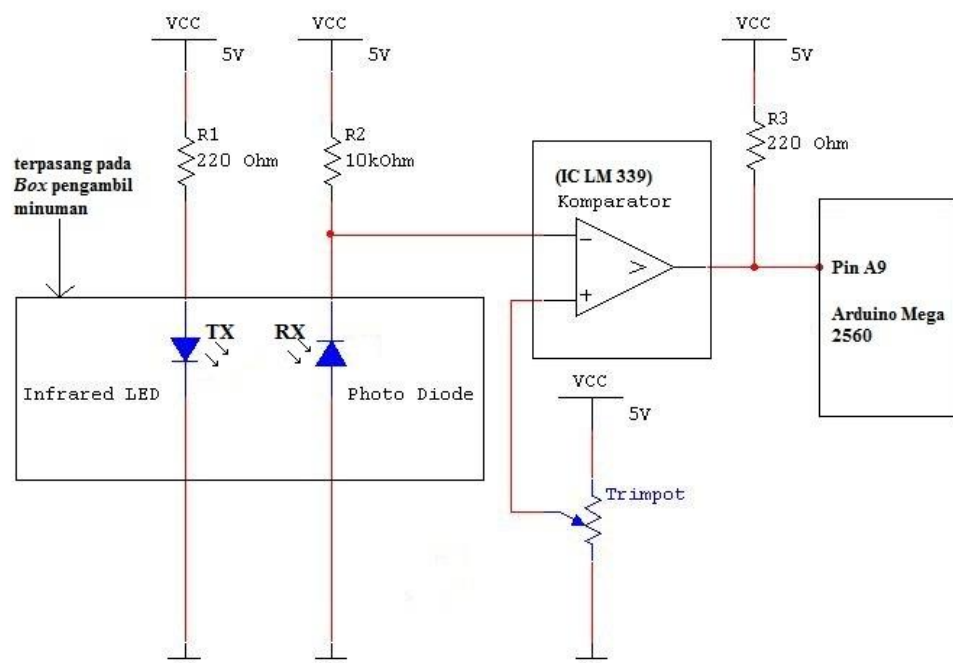
Motor DC yang digunakan adalah motor DC yang beroperasi dengan tegangan 9 Volt. Untuk driver motor DC menggunakan IC L293D. Motor DC pada alat digunakan sebagai *actuator* sumbu Z, *actuator conveyor*, *actuator* pintu tempat pengambilan minuman. Gambar 3.8 menjelaskan rancangan skematik pengendali motor DC yang digunakan pada *vending machine*.



Gambar 3.8 Skematik Actuator Sumbu X dan Sumbu Y (Motor DC)

4) Rangkaian Sensor Cahaya

Sensor cahaya digunakan sebagai pendeteksi keberadaan minuman kaleng dalam *box* tempat pengambilan minuman dan *storage* tempat penyimpanan minuman. Sensor Cahaya menggunakan komponen utama IC LM 339 sebagai komparator, photodiode sebagai *receiver*, dan LED sebagai *transmitter*. Gambar 3.9 menjelaskan rancangan skematik sensor cahaya yang digunakan pada *vending machine*.



Gambar 3.9 Skematik Sensor Cahaya (pada box tempat pengambilan minuman)

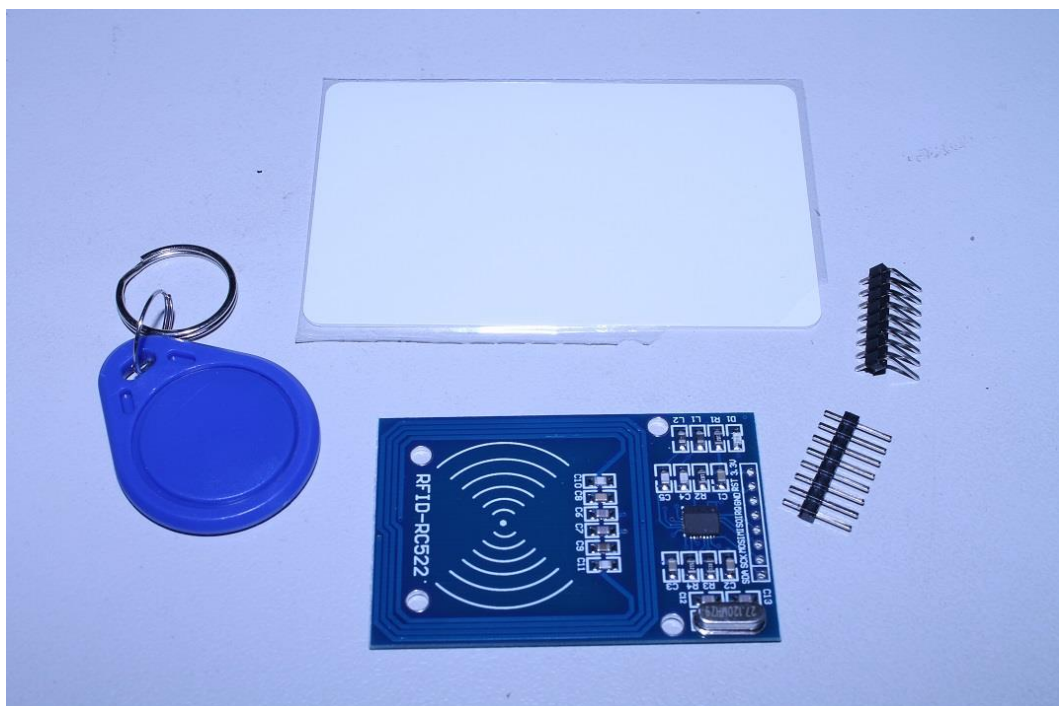
Skematik yang serupa juga digunakan pada pendeteksi persediaan minuman kaleng pada *storage*, tabel 3.1 menjelaskan pin Arduino Mega 2560 yang digunakan untuk sensor *box* dan sensor pada *storage*.

Tabel 3.1 Daftar Penggunaan Pin Arduino Mega 2560 Untuk Sensor Cahaya

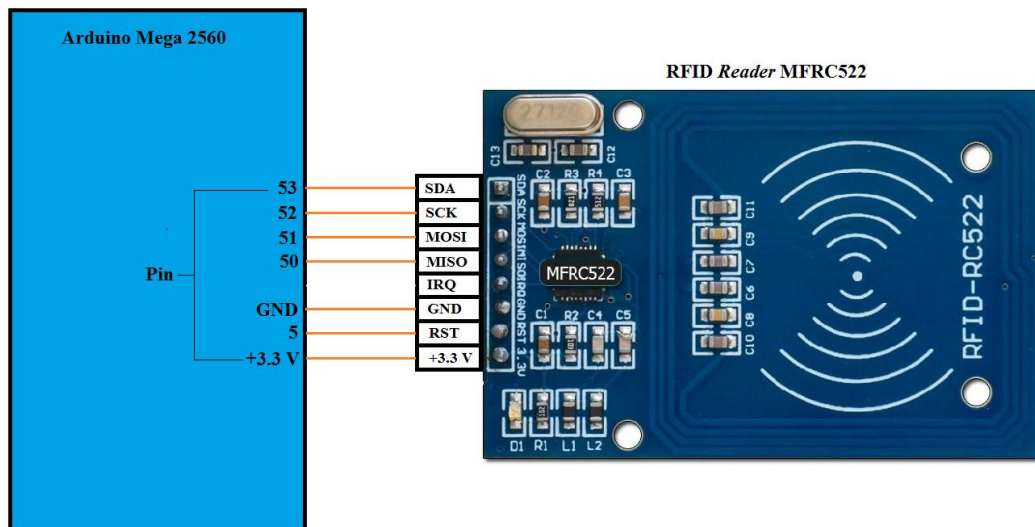
Fungsi Sensor	Pin Arduino
Pendeteksi minuman kaleng pada <i>box</i>	A9
Pendeteksi persediaan minuman kaleng pada <i>storage</i> 1	31
Pendeteksi persediaan minuman kaleng pada <i>storage</i> 2	32
Pendeteksi persediaan minuman kaleng pada <i>storage</i> 3	33
Pendeteksi persediaan minuman kaleng pada <i>storage</i> 4	34
Pendeteksi persediaan minuman kaleng pada <i>storage</i> 5	35
Pendeteksi persediaan minuman kaleng pada <i>storage</i> 6	36
Pendeteksi persediaan minuman kaleng pada <i>storage</i> 7	37
Pendeteksi persediaan minuman kaleng pada <i>storage</i> 8	38
Pendeteksi persediaan minuman kaleng pada <i>storage</i> 9	39

5) Modul RFID Reader MFRC522

Untuk pembaca kartu *E-Money* pada *vending machine* menggunakan modul RFID *reader* MFRC522 dan Kartu RFID MIFARE yang beroperasi pada frekuensi 13,56 MHz. Modul RFID *reader* MFRC522 dan MIFARE *cards* dapat dilihat pada gambar 3.10, lalu Skematik rangkaian terlihat pada gambar.3.11



Gambar 3.10 Modul RFID Reader MFRC522 dan MIFARE Cards



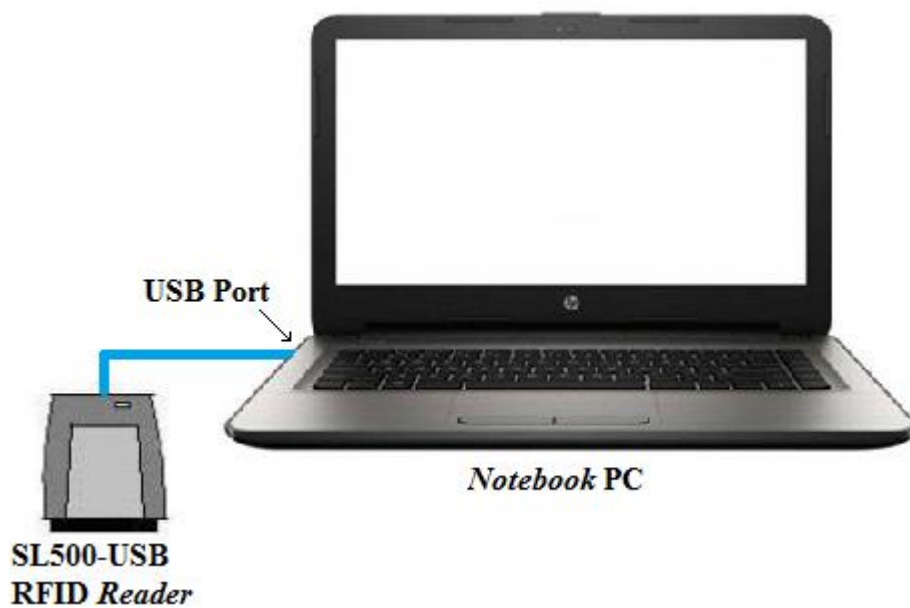
Gambar 3.11 Skematik RFID Reader

6) SL500-USB (RFID reader plug and play)

Administrator menggunakan modul RFID reader SL500-USB sebagai pembaca kartu e-money (MIFARE cards). Cara instalasi modul ini dengan menghubungkan modul dengan PC menggunakan kabel USB. MIFARE cards akan terbaca sebagai kode 10 digit decimal dan langsung terbaca pada kursor yang ditunjuk pada aplikasi software yang sedang dibuka di *notebook* PC (dalam hal ini aplikasi yang digunakan adalah software aplikasi administrator e-money). Modul SL500-USB dapat dilihat pada gambar 3.12, dan instalasi pemasangan modul SL500-USB pada PC diperlihatkan pada gambar 3.13.



Gambar 3.12 Modul RFID USB Plug and Play

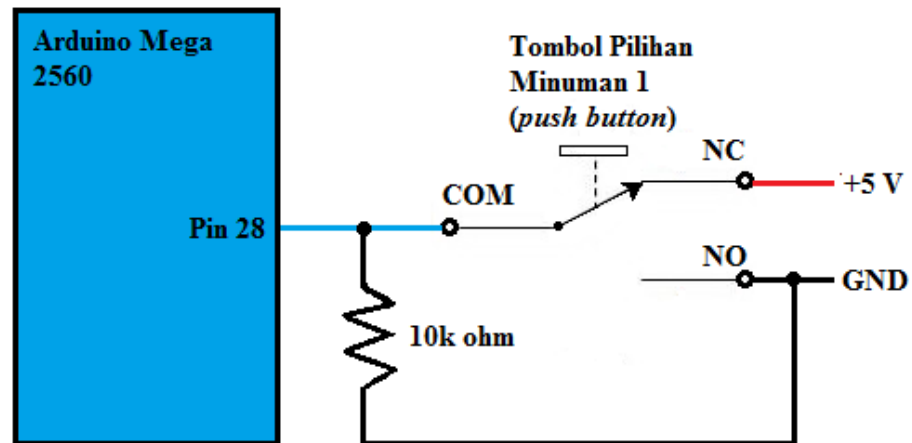


Gambar 3.13 Instalasi Modul SL500-USB (RFID reader plug and play) ke Notebook PC

7) Tombol Pilihan Minuman Kaleng (*push button*)

Untuk memilih minuman kaleng yang akan dibeli, maka diperlukan tombol pilihan minuman kaleng. Tombol pilihan minuman kaleng dirancang menggunakan *push button* dan diberikan resistor 10k ohm pada COM sebagai

pull-down. Gambar 3.14 merupakan rancangan skematik tombol pilihan minuman kaleng yang dihubungkan ke Arduino Mega 2560.



Gambar 3.14 Rancangan Skematik Tombol Pilihan Minuman Kaleng

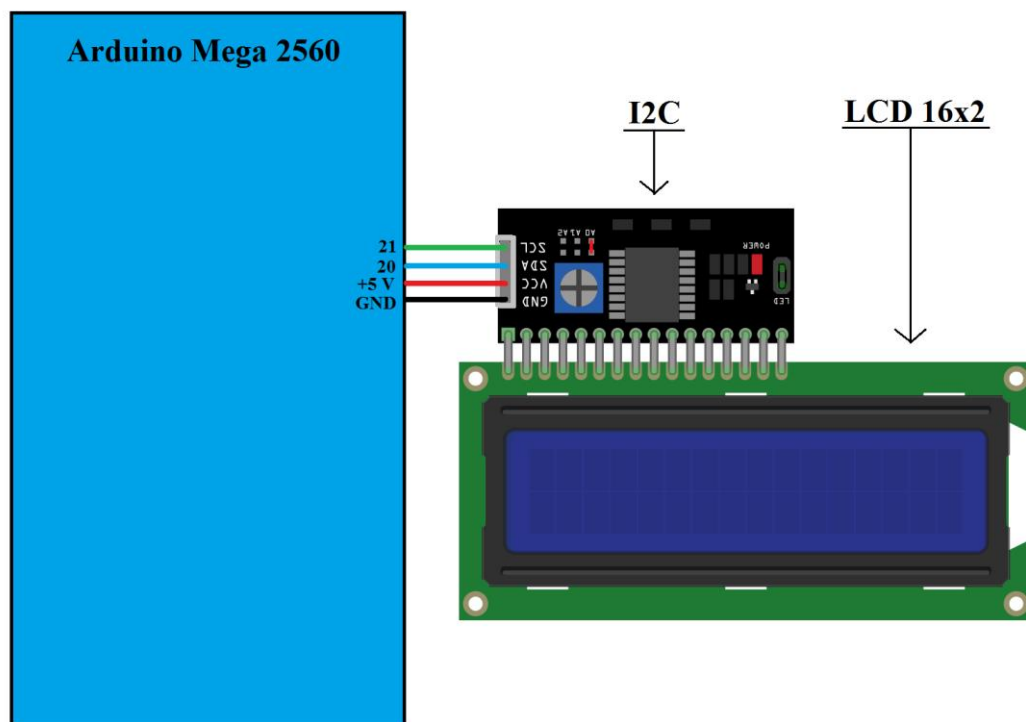
Skematik pada gambar 3.14 merupakan skematik yang digunakan tombol pilihan minuman kaleng nomor 1, skematik yang serupa juga diterapkan pada tombol lainnya. Tabel 3.2 menjelaskan hubungan setiap rangkaian tombol dengan Arduino Mega 2560.

Tabel 3.2 Hubungan Rangkaian Tombol dengan Arduino Mega 2560

Nama Tombol	Pin Arduino Mega 2560 yang digunakan
YES	40
NO	41
1	28
2	29
3	30
4	25
5	26
6	27
7	22
8	23
9	24

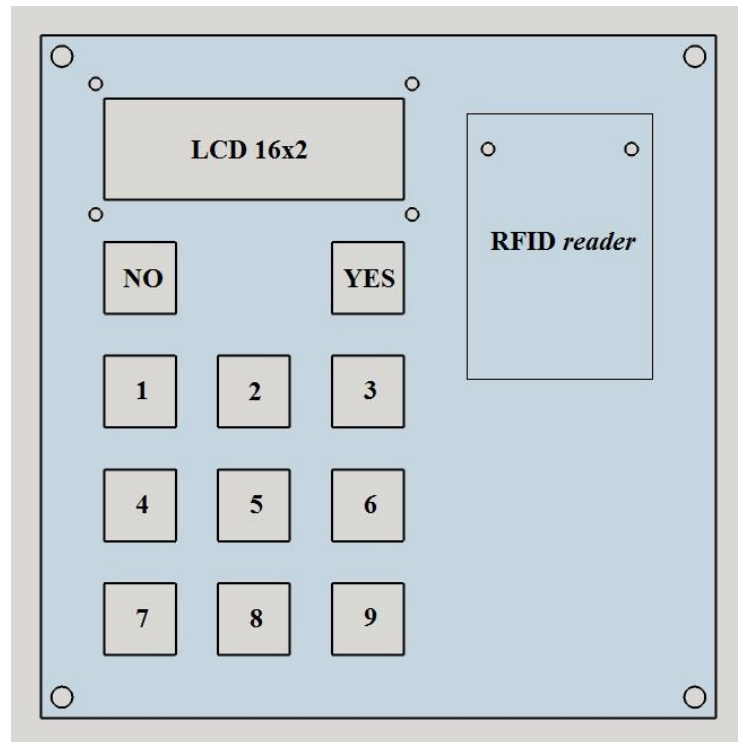
8) Text LCD 16x2 dengan I2C Backpack

Tampilan monitor antarmuka (*interface*) dibutuhkan untuk interaksi *vending machine* dengan pembeli minuman kaleng. Tampilan *interface vending machine* ini dirancang menggunakan Text LCD 16x2. Modul Text LCD 16x2 dipasangkan dengan I2C untuk meminimalisir jumlah pin Arduino Mega 2560. Rancangan skematik Text LCD 16x2 I2C dapat dilihat pada gambar 3.15.



Gambar 3.15 Skematik LCD 16x2 I2C ke Arduino Mega 2560

Setiap komponen *interface* dalam *vending machine* ditempatkan dalam satu panel *interface*. Komponen interface tersebut terdiri dari Text LCD 16x2 I2C, tombol-tombol pilihan minuman kaleng (*push button*), dan RFID *reader*. Rancangan panel *interface* dapat dilihat pada gambar 3.16.



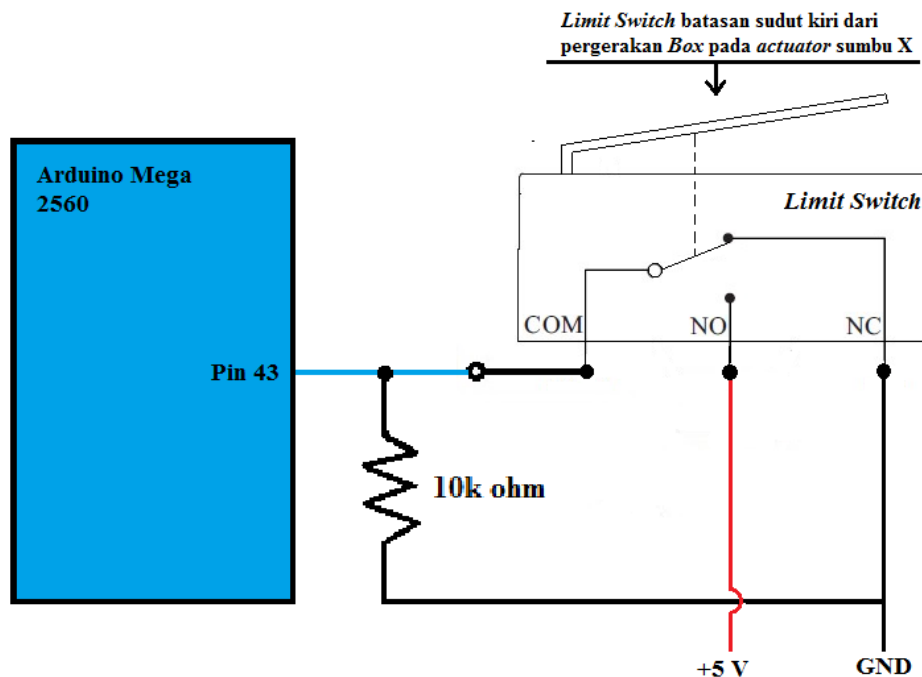
Gambar 3.16 Rancangan Panel Interface

9) Pendeteksi Batas Pergerakan *Actuator*

Limit Switch pada alat digunakan untuk pembatasan pergerakan *actuator*, yakni membatasi pergerakan buka-tutup pintu tempat pengambilan minuman, batas depan-belakang *actuator* sumbu Z, dan mendeteksi posisi awalan *box* pengambilan minuman (batas bawah sumbu Y dan kiri sumbu X). Bentuk fisik *Limit switch* dapat dilihat pada gambar 3.17. dan skematik *limit switch* sebagai pendeteksi batas pergerakan *Box* pada *actuator* dapat dilihat pada gambar 3.18.



Gambar 3.17 Limit Switch



Gambar 3.18 Rancangan Skematik *Limit Switch* Sebagai Pendeteksi Batasan Pergerakan *Actuator*

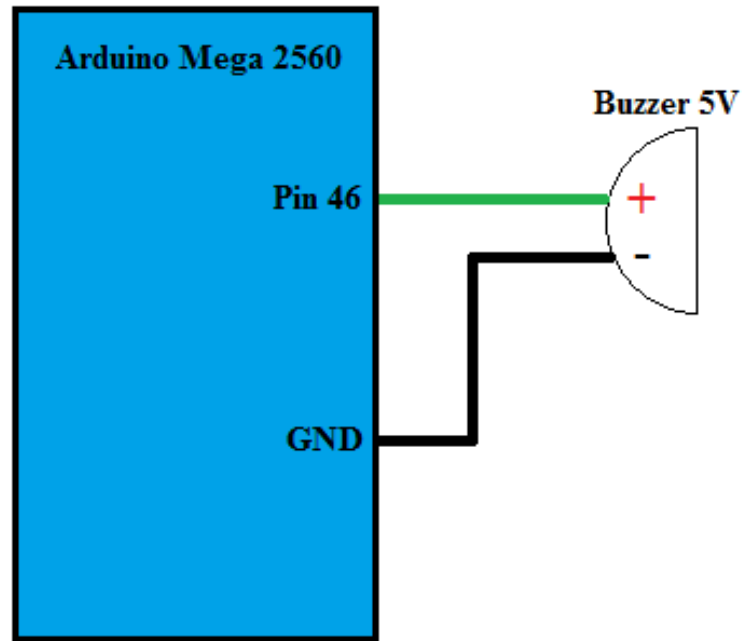
Gambar 3.16 merupakan salah satu skematik yang menggunakan *limit switch* sebagai pendeteksi batasan pergerakan *actuator*, selain itu skematik yang serupa gambar tersebut juga diterapkan pada pendeteksi batasan pergerakan *actuator* lainnya dalam sistem, hal tersebut dijelaskan dalam tabel 3.3 tentang hubungan rangkaian *limit switch* dengan pin Arduino Mega 2560.

Tabel 3.3 Hubungan Rangkaian *Limit Switch* dengan Arduino Mega 2560

Nama Komponen	Pin Arduino Mega 2560
<i>Limit Switch</i> Kiri	43
<i>Limit Switch</i> Bawah	45
<i>Limit Switch</i> Sumbu Z Depan	A6
<i>Limit Switch</i> Sumbu Z Belakang	A7
<i>Limit Switch</i> Pintu Buka	A12
<i>Limit Switch</i> Pintu Tutup	A13

10) Buzzer

Buzzer dalam *vending machine* ini digunakan sebagai tanda peringatan dalam bentuk suara. Buzzer dipasang ke pin 46 Arduino Mega 2560. Skematik rangkaian buzzer dapat dilihat pada gambar 3.19.



Gambar 3.19 Skematik Buzzer

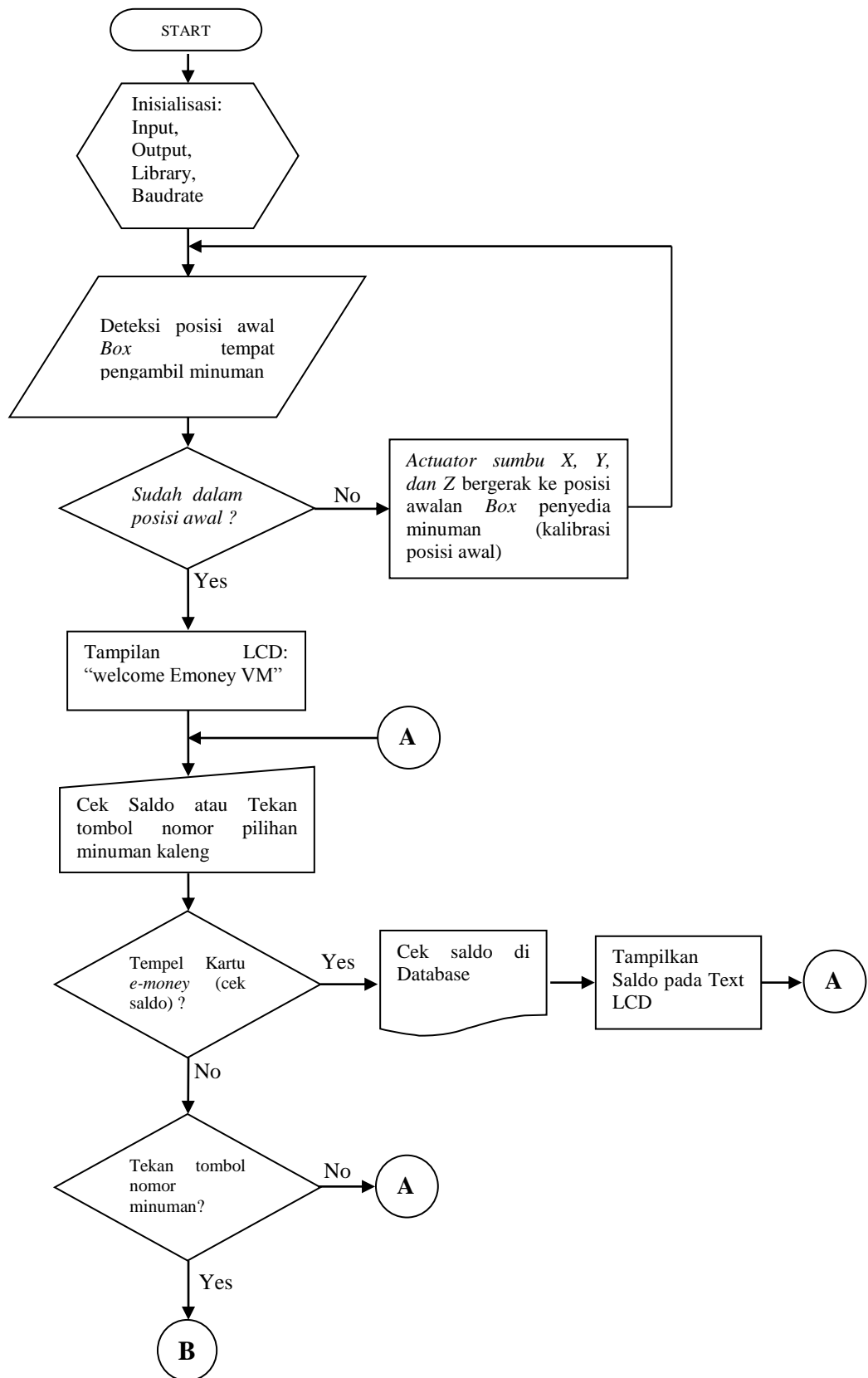
3.6. Rancangan Siklus Kerja *Vending Machine* dan Program Arduino IDE

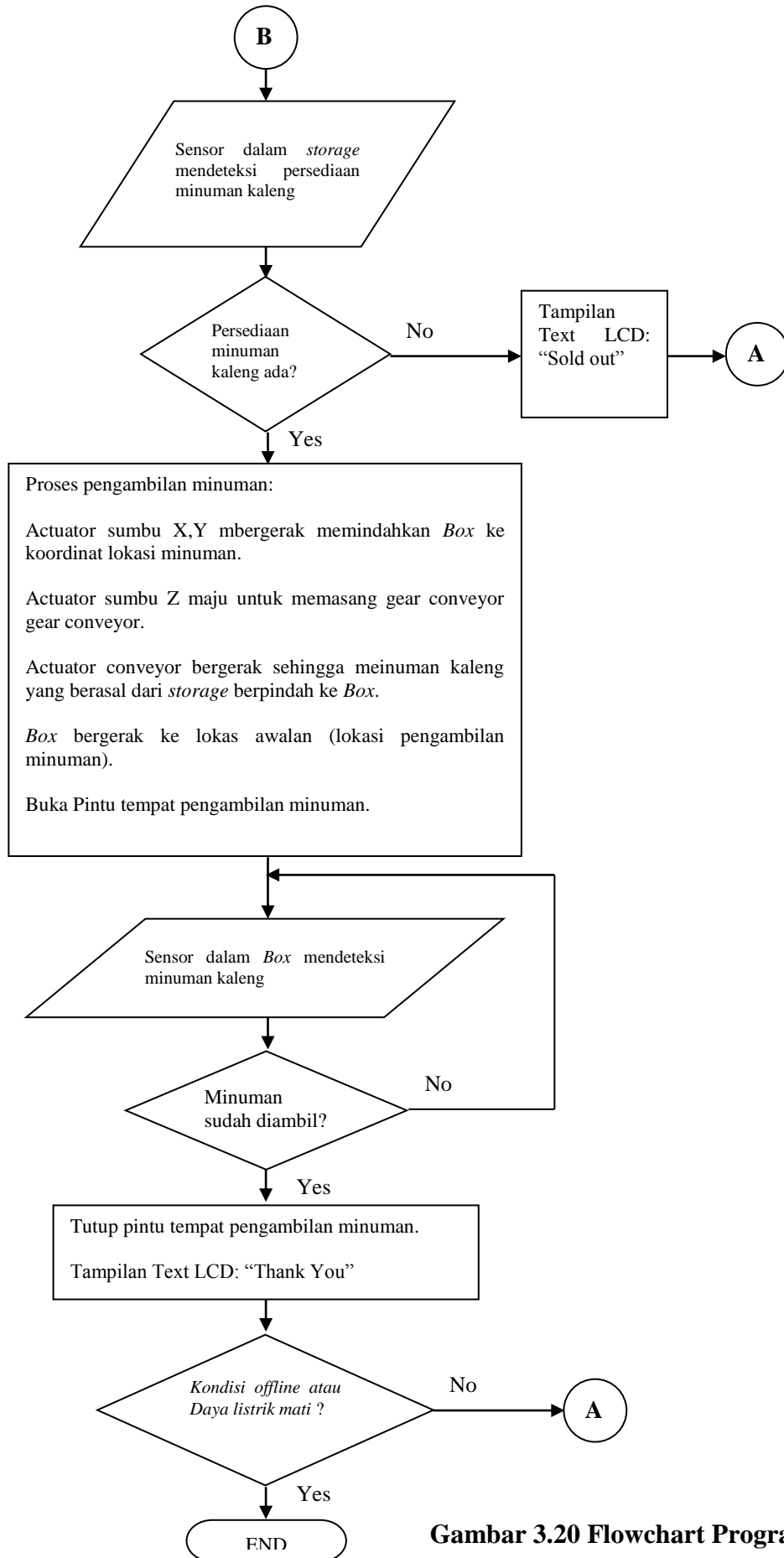
Secara umum, cara kerja *vending machine* minuman kaleng dapat dijelaskan melalui penjelasan siklus kerja *vending machine* berikut ini:

- 1) Hidupkan power supply *vending machine* dan PC
- 2) Buka aplikasi *Emoney_database.exe* di PC, lalu hubungkan dengan *vending machine*
- 3) Pembeli datang ke administrator untuk membeli kartu *E-Money*
- 4) Administrator mendaftarkan kartu dan isi saldo melalui aplikasi *Emoney_database.exe*, lalu memberikan kartu tersebut ke pembeli
- 5) Pembeli datang ke *vending machine* untuk membeli minuman
- 6) Tampilan awal LCD 16x2 pada *vending machine* adalah “Welcome to *Emoney VM*”, apabila kartu *E-Money* ditempelkan pada *E-Money reader* (RFID reader) *vending machine* maka *text LCD vending machine* akan menampilkan isi saldo kartu tersebut.
- 7) Apabila tombol pilihan minuman ditekan, maka:
 - a) *Text LCD* pada *vending machine* akan menampilkan “Sorry ... SOLD OUT !!” jika persediaan minuman yang dipilih tidak tersedia
 - b) Jika persediaan minuman yang dipilih tersedia, maka *vending machine* akan menanyakan melalui tampilan LCD: “ARE YOU SURE ? NO / YES”, jika tekan tombol NO maka program akan kembali ke tampilan LCD 16x2 di awal, sedangkan jika tekan tombol YES maka *text LCD* akan menampilkan “Please TAP YOUR CARD”

- 8) Setelah kartu ditempelkan, maka *box* tempat pengambilan minuman akan bergerak menuju koordinat minuman kaleng yang dipilih, kemudian *conveyor* pada *storage* minuman kaleng akan bergerak sehingga minuman kaleng tersebut berpindah ke dalam *box* tempat pengambilan minuman. Setelah *box* tempat pengambilan minuman terisi minuman kaleng, maka *box* tempat pengambilan minuman akan bergerak menuju tempat pengambilan minuman dan pintu akan terbuka sehingga pembeli dapat mengambil minuman tersebut.
- 9) Setelah minuman diambil oleh pembeli, maka pintu akan tertutup dan *text* LCD akan menampilkan “*Thank You*”.

Software Arduino IDE digunakan untuk memprogram *board* Arduino Mega 2560 untuk melakukan fungsinya dalam sistem. Berdasarkan rancangan siklus kerja *vending machine* secara umum yang telah dijelaskan, maka rancangan *flowchart vending machine* untuk program Arduino Mega 2560 dapat dilihat pada gambar 3.20.

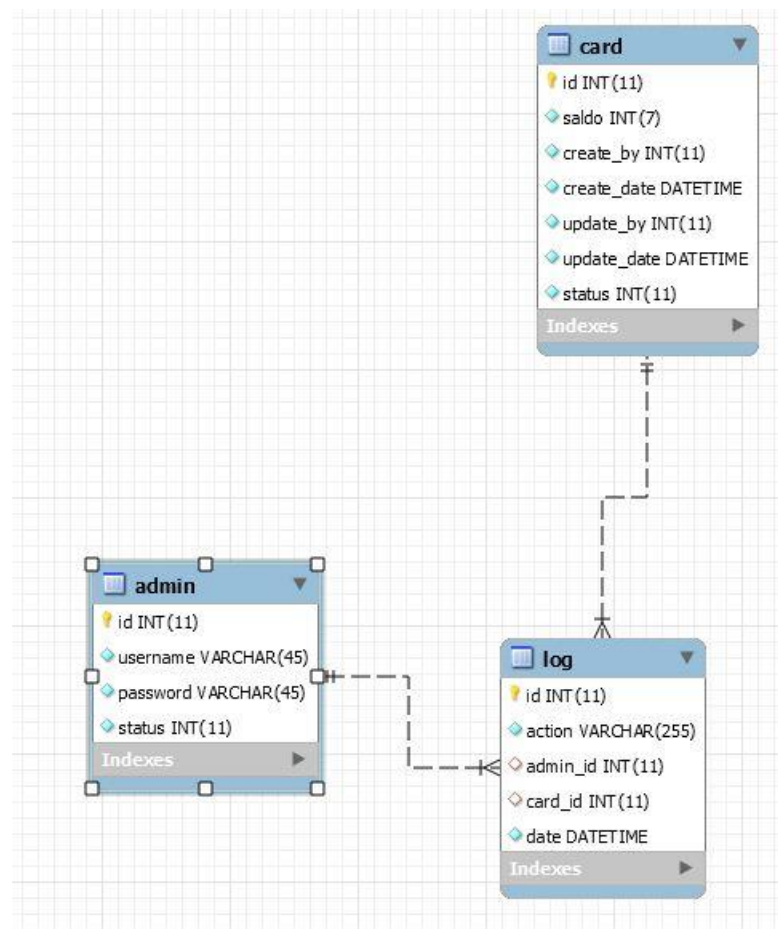




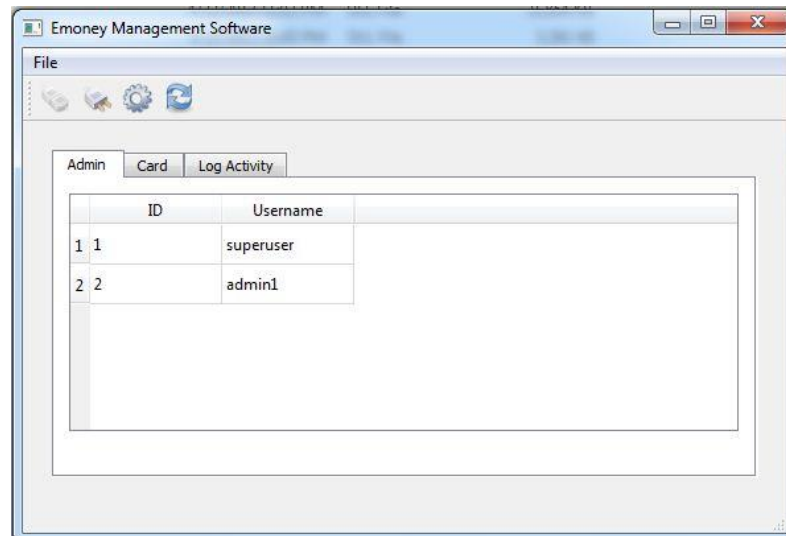
Gambar 3.20 Flowchart Program

3.7. Rancangan Sistem *E-Money* (database dan aplikasi administrator)

Database *E-Money* dibuat menggunakan MySQL Workbench, dan aplikasi *interface* administrator menggunakan QT Creator. Rancangan *database* dapat dilihat pada gambar 3.21. Sedangkan tampilan *interface* administrator *E-Money* dapat dilihat pada gambar 3.22.



Gambar 3.21 Rancangan EER Diagram database *E-Money* pada MySQL Workbench



Gambar 3.22 Rancangan tampilan interface program Emoney_Database.exe yang dibuat menggunakan Qt Creator

3.8. Teknik dan Prosedur Pengumpulan Data

Pengumpulan data dilakukan dengan tujuan untuk mendapatkan data dan informasi yang dibutuhkan. Pengumpulan data yang dibutuhkan adalah data pulsa yang dibutuhkan motor stepper sumbu X dan Sumbu Y, data koordinat lokasi setiap storage minuman kaleng, pembacaan kartu RFID, simulasi transaksi *E-Money*, dan waktu pelayanan penyediaan minuman.

Agar *actuator* agar mencapai koordinat *storage* yang sesuai maka dibutuhkan pengambilan data pulsa step yang dibutuhkan. Data ini bisa didapatkan dengan melakukan uji coba secara langsung menggerakkan *actuator* sumbu X dan Y sampai mendapatkan lokasi yang tepat lalu dicatat hasilnya.

Proses transaksi *E-Money* diuji coba dengan melakukan simulasi pembelian minuman kaleng, lalu hasilnya dicatat. Sistem *vending machine* secara keseluruhan diuji coba dengan simulasi dan dicatat juga berapa waktu pelayanan yang dibutuhkan.

BAB IV

HASIL PENELITIAN DAN PEMBAHASAN

4.1. Deskripsi Hasil Penelitian

Hasil dari penelitian ini yaitu *vending machine* minuman kaleng berbasis Arduino Mega 2560 yang menggunakan *E-money* sebagai alat transaksi pembelian minuman kaleng. Prototipe mesin ini menyediakan 9 macam jenis minuman kaleng ukuran 330 ml yang setiap jenisnya tersimpan pada *storage*. Setiap *storage* dapat menyimpan maksimal 3 minuman kaleng, jadi total maksimal minuman kaleng yang dapat di simpan dalam *vending machine* ini berjumlah 27 minuman kaleng.

Storage berbentuk *conveyor* tanpa *actuator*, tapi terdapat *gear* yang nantinya akan digerakkan oleh *actuator* (motor DC) yang menyatu/terpasang pada kotak pengambilan minuman kaleng (*box*). Minuman kaleng yang berdidri diatas *conveyor* akan digerakkankan seiring pergerakan *actuator* yang telah menempel pada *conveyor* lalu berpindah posisi masuk ke dalam *box*. Terdapat *actuator* sumbu X (motor stepper), Y (motor stepper), dan Z (motor DC) untuk menggerakkan *box* tempat pengambilan minuman agar dapat mencapai setiap koordinat lokasi minuman kaleng. Pembeli nantinya akan mengambil minuman kaleng di dalam *box* pengambilan minuman. Pada lokasi awal *box* terdapat mekanik pintu yang dapat terbuka dan tertutup yang digunakan sebagai pintu tempat pengambilan minuman kaleng untuk pembeli.

Panel *interface* pada *vending machine* tersedia sebagai media komunikasi/interaksi antara mesin dengan pembeli. Panel *interface* terdidri dari

tombol-tombol nomor pilihan minuman kaleng, tombol YES, tombol NO, pembaca kartu *E-Money* (RFID reader), dan LCD 16x2.

Untuk database *E-Money* menggunakan database MySQL Workbench. Data informasi identitas (ID)/ kode kartu *E-Money*, saldo kartu, dan waktu transaksi tersimpan dalam database. Software aplikasi *Emoney_database.exe* digunakan oleh administrator *E-Money* untuk mendaftarkan kartu terhadap database sistem dan mengisi nilai saldo untuk kartu *E-Money*. Kartu *E-Money* (MIFARE cards) harus di daftarkan oleh administrator ke dalam sistem database melalui aplikasi sebelum diberikan kepada pembeli kartu (calon pembeli minuman kaleng pada *vending machine*), jika tidak terdaftar maka kartu tidak berlaku untuk alat bayar. Kartu *E-Money* dibeli melalui administrator *E-money*. Hanya pembeli yang memiliki kartu *E-Money* yang sudah terdaftar yang dapat membeli minuman kaleng.

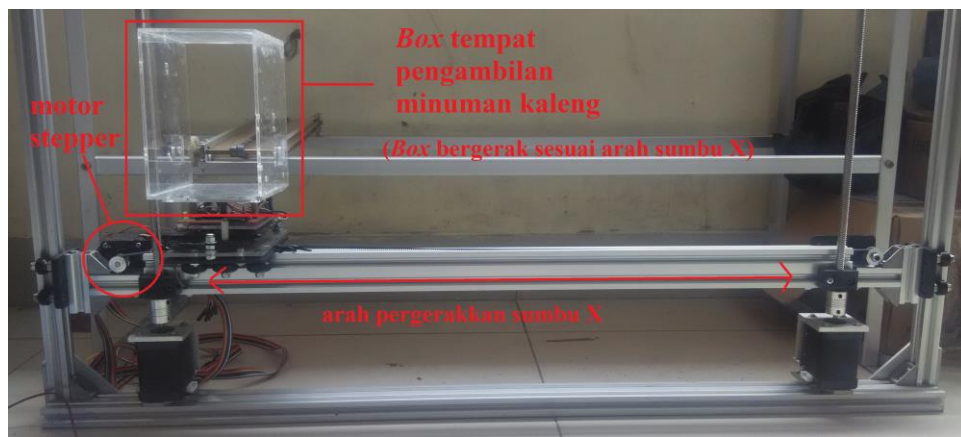
Pembelian minuman kaleng berlaku satu kali untuk setiap transaksi saat penempelan kartu. Setelah pembelian maka saldo akan berkurang sesuai harga yang tertera pada unit *vending machine*. Harga minuman nomor 1, 2, dan 3 adalah Rp 5.000; harga minuman nomor 4, 5, dan 6 adalah Rp 8.000; harga minuman nomor 7, 8, dan 9 adalah Rp 10.000.

Penjelasan hasil rancangan mekanik, elektronik, dan *software* serta cara kerja alat dijelaskan pada 4.1.1., 4.1.2., dan 4.1.3.

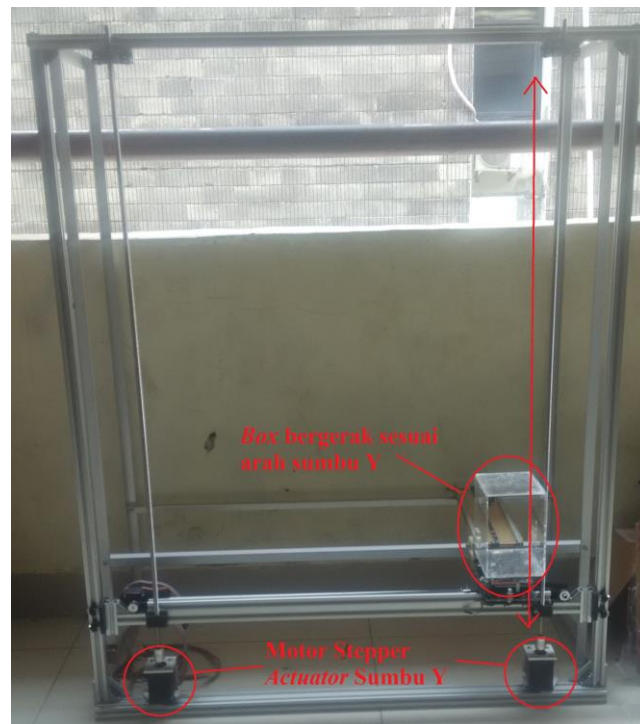
4.1.1. Hasil Rancangan Mekanik

Struktur penggerak utama dalam alat ini terdiri dari mekanik untuk pergerakan linear arah sumbu X-Y-Z, mekanik *Box* tempat pengambilan minuman, mekanik pintu buka-tutup tempat pengambilan minuman, dan 9 buah

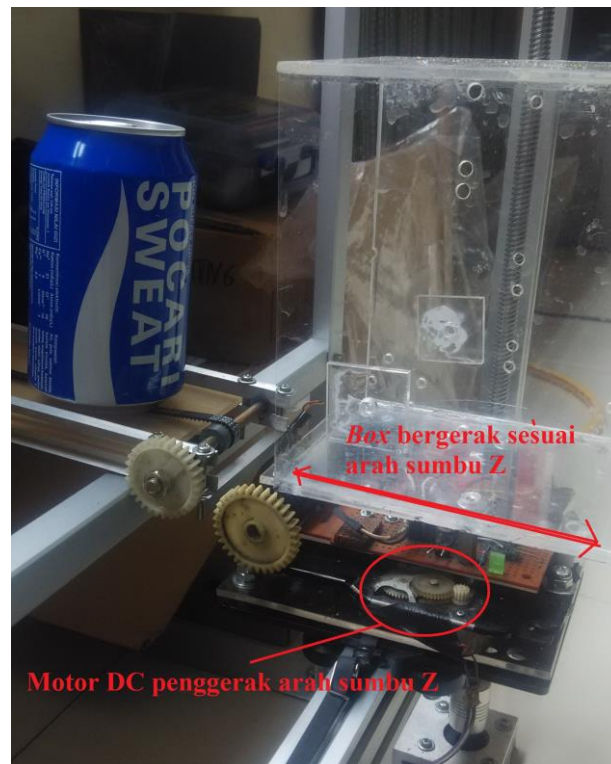
storage tempat persediaan minuman dalam bentuk *conveyor* (tanpa motor DC, karena motor DC terpasang pada *Box* tempat pengambilan minuman). Hasil pembuatan mekanik sumbu X dapat dilihat pada gambar 4.1, mekanik sumbu Y pada gambar 4.2, sumbu Z pada gambar 4.3, mekanik *Box* tempat pengambilan minuman pada gambar 4.4, mekanik pintu pada gambar 4.5, dan mekanik *storage* (*conveyor*) pada gambar 4.6.



Gambar 4.1 Mekanik Actuator Sumbu X



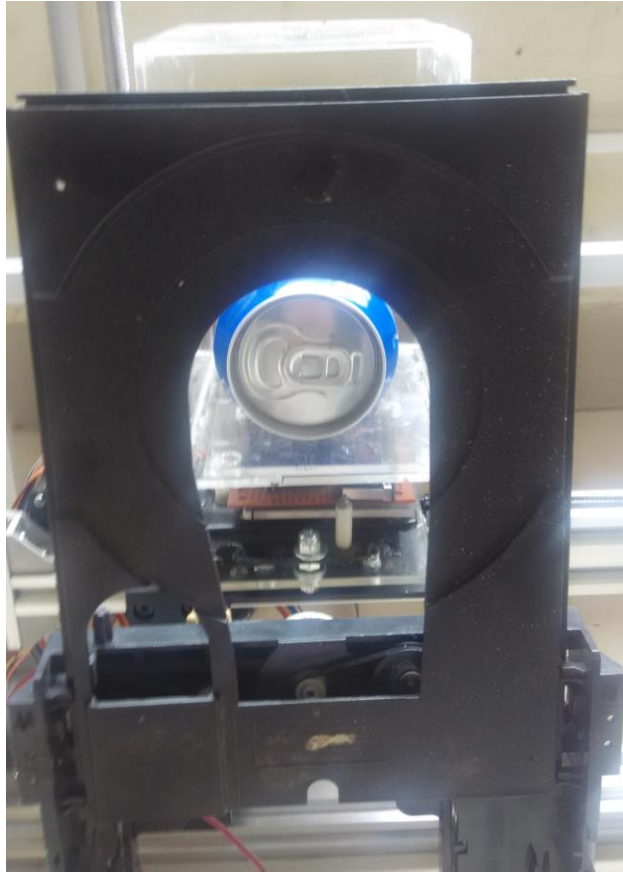
Gambar 4.2 Mekanik Actuator Sumbu Y



Gambar 4.3 Mekanik Actuator Sumbu Z



Gambar 4.4 *Box* tempat pengambilan minuman



Gambar 4.5 Mekanik Pintu Tempat Pengambilan Minuman Kaleng



Gambar 4.6 Mekanik *Conveyor* Storage (tempat persediaan minuman)

4.1.2. Hasil Rancangan Elektronik

Semua Rangkaian Elektronik terhubung dengan Arduino Mega 2560 sebagai pusat proses sistem. Table 4.1 menjelaskan tentang hubungan pin Arduino Mega 2560 dengan semua perangkat pendukung sistem, sedangkan untuk skematik utuh dan hasil rangkaian terdapat pada lampiran.

Tabel 4.1 Hubungan Pin Rangkaian Elektronik

No	Perangkat Elektronik	Nama Pin Perangkat Elektronik	Nomor Pin pada Arduino Mega 2560
1.	RFID Reader MFR522	RST	5
		SDA	53
		MOSI	51
		MISO	50
		SCK	52
2.	Panel Tombol	Button 1	28
		Button 2	29
		Button 3	30
		Button 4	25
		Button 5	26
		Button 6	27
		Button 7	22
		Button 8	23
		Button 9	24
		Button Yes	40
		Button No	41
3.	Modul Sensor Cahaya	Storage 1	31
		Storage 2	32
		Storage 3	33
		Storage 4	34
		Storage 5	35
		Storage 6	36
		Storage 7	37
		Storage 8	38
		Storage 9	39
		Box	A9
4.	Text LCD 16x2 dengan <i>backpack</i> I2C	SDA	20
		SCL	21
5.	Driver Motor Stepper Sumbu X	Step	6
		Dir	7
6.	Driver Motor Stepper Sumbu Y	Step	8
		Dir	9

7.	Driver Motor DC Sumbu Z	DirFront	A4
		DirBack	A5
8.	<i>Conveyor</i>	DirCCW	A8
9.	Driver Motor DC Pintu Pengambilan Minuman	DirOpen	A10
		DirClose	A11
10.	Sensor Mekanik Limit Switch	L_Left	43
		L_Down	45
		L_Front	A6
		L_Back	A7
		L_Open	A12
		L_Close	A13
11	Buzzer	+	46

4.1.3. Hasil Rancangan Software

Software yang dibuat meliputi program *Sketch* yang dibuat dan diupload dari software Arduino IDE pada komputer menuju ke *board* Arduino Mega2560. Isi program *Sketch* dirancang sesuai flowchart. Gambar 4.7. memperlihatkan *Sketch* yg dibuat dengan Arduino IDE, sedangkan source program selengkapnya dapat dilihat pada lampiran.

```

Emoney_VendingMachine | Arduino 1.6.7
File Edit Sketch Tools Help
Emoney_VendingMachine
15 #include <Wire.h>
16 #include <LiquidCrystal_I2C.h>
17
18 #include <SPI.h>
19 #include <MFRC522.h>
20
21 #define SS_PIN 53
22 #define RST_PIN 5
23
24 LiquidCrystal_I2C lcd(0x3F, 16, 2);
25
26 MFRC522 rfid(SS_PIN, RST_PIN); // Instance of the class
27
28 MFRC522::MIFARE_Key key;
29
30 // Init array that will store new NUID
31 byte nuidPICC[3];
32
33 // Tombol Pilihan Minuman
34 const int pb_btn[9] = {
35   28, 29, 30,
36   25, 26, 27,
37   22, 23, 24
38 };
39 const int harga[9] = {
40   5000, 5000, 5000,
41   8000, 8000, 8000,
42   10000, 10000, 10000

```

Gambar 4.7 Tampilan program *Sketch* Arduino yang telah dibuat di Software Arduino IDE

Software selanjutnya yang dibuat adalah database *E-Money* yang dibuat menggunakan MySQL Workbench. Database *E-Money* yang dibuat terdiri dari tabel admin, tabel card, dan tabel log.

1) Tabel Admin

Tabel admin berisi kolom ID, username, password, dan status. ID merupakan nomor urut identitas seorang administrator *E-Money*. Username merupakan nama administrator yang terdaftar di database yang

digunakan saat login, kemudian password merupakan kode rahasia yang disimpan oleh administrator saat proses login. Kolom status merupakan tanda keaktifan administrator/user, jika berisi 1 maka tandanya aktif, sedangkan jika 0 maka tandanya administrator sedang dalam kondisi nonaktif. Gambar 4.8 merupakan *printscreen* database pada table admin.

	id	username	password	status
▶	1	superuser	superuser	1
	2	admin1	admin1	1
*	NULL	NULL	NULL	NULL

Gambar 4.8 Tabel Admin pada database *E-Money*

2) Tabel Card

Tabel admin berisi kolom id, saldo, create_by, create_date, update_by, update_date, dan status. id pada table card merupakan kode unik yang terdapat pada kartu *E-Money*. Kolom saldo berisi nilai uang dalam rupiah, kolom create_by berisi kode id administrator yang membuat data di waktu tersebut. Kolom create_date berisi waktu saat data dibuat. Kolom update_by berisi kode id administrator yang mengubah data.. Kolom update_date berisi waktu saat data diubah. Kolom status merupakan kode kartu *E-Money* saat status aktif (1) dan nonaktif (0). Gambar 4.9 merupakan *printscreen* database pada table card.

	id	saldo	create_by	create_date	update_by	update_date	status
▶	724427715	120000	2	2017-06-14 14:33:58	2	2017-06-14 14:33:58	1
	1038523004	14000	2	2017-06-11 07:00:50	2	2017-06-11 07:00:50	1
	1390740421	10000	2	2017-07-30 07:35:43	2	2017-07-30 07:35:43	1
	2058879838	55000	2	2017-06-11 06:59:46	2	2017-06-11 06:59:46	1
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Gambar 4.9 Database pada Table Card

3) Tabel Log

Tabel log berisi kolom id, action, admin_id, card_id, dan date. id pada table log merupakan nomor urut transaksi yang terjadi. Kolom action berisi rekam jejak aksi yang dilakukan. Kolom admin_id berisi nomer kode id administrator yang sedang melakukan aktifitas/aksi tersebut. Kolom card_id berisi kode unik id pada kartu *E-Money*. Kolom date berisi tanggal saat aksi dilakukan. Gambar 4.10 merupakan *printscreen* database pada table log.

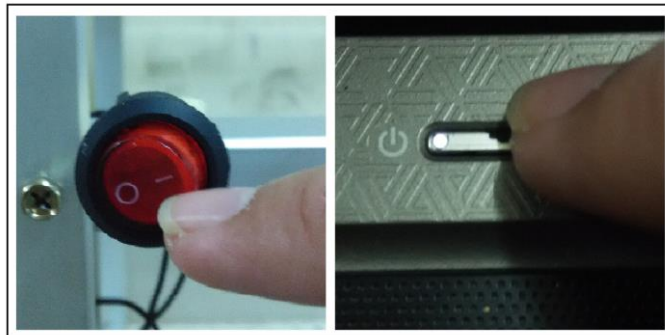
	id	action	admin_id	card_id	date
	15	Berhasil membuat operator baru dengan Username = admin1	1	NULL	2017-06-11 06:58:38
▶	16	Berhasil Menambah Card Baru dengan ID = 2058879838, Saldo = 20000	2	NULL	2017-06-11 06:59:46
	17	Berhasil menambahkan saldo sebesar 10000 ke RFID = 2058879838	2	NULL	2017-06-11 07:00:10
	18	Berhasil Menambah Card Baru dengan ID = 1038523004, Saldo = 50000	2	NULL	2017-06-11 07:00:50
	19	Berhasil menambahkan saldo sebesar 10000 ke RFID = 1038523004	2	NULL	2017-06-11 07:01:46
	20	Berhasil Menambah Card Baru dengan ID = 724427715, Saldo = 10000	2	NULL	2017-06-14 14:33:58
	21	Berhasil menambahkan saldo sebesar 50000 ke RFID = 724427715	2	NULL	2017-06-14 14:34:43
	22	Berhasil menambahkan saldo sebesar 20000 ke RFID = 724427715	2	NULL	2017-07-04 10:03:13
	23	Berhasil menambahkan saldo sebesar 100000 ke RFID = 2058879838	2	NULL	2017-07-24 03:20:58
	24	Berhasil menambahkan saldo sebesar 100000 ke RFID = 1038523004	2	NULL	2017-07-24 03:21:12
	25	Berhasil menambahkan saldo sebesar 100000 ke RFID = 724427715	2	NULL	2017-07-24 03:21:23
	26	Berhasil menambahkan saldo sebesar 100000 ke RFID = 724427715	2	NULL	2017-07-30 07:11:51
	27	Berhasil menambahkan saldo sebesar 50000 ke RFID = 2058879838	2	NULL	2017-07-30 07:22:20
	28	Berhasil Menambah Card Baru dengan ID = 1390740421, Saldo = 10000	2	NULL	2017-07-30 07:35:43
*	NULL	NULL	NULL	NULL	NULL

Gambar 4.10 Database Pada Table Log

4.1.4. Hasil Pembuatan Sistem Kerja Alat

Rancangan sistem kerja *Vending machine* diujicoba fungsinya melalui simulasi. Berikut ini adalah penjelasan sistem kerja alat berdasarkan simulasi yang telah dilakukan.

- 1) Aktifkan power supply *vending machine* dan PC (gambar 4.11)

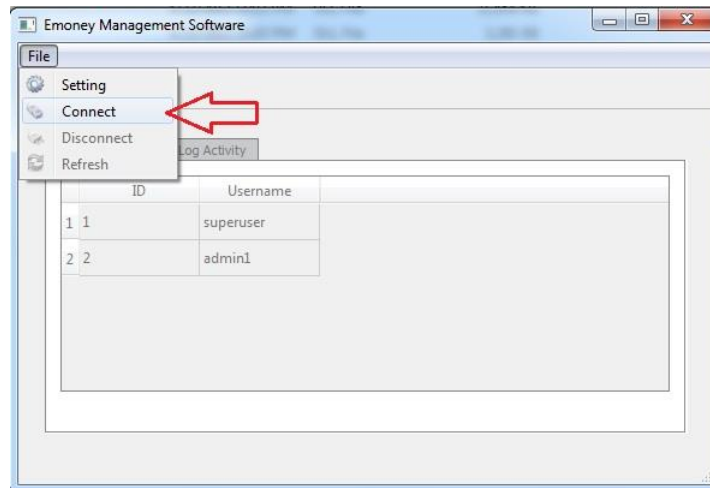


Gambar 4.11 Tombol Power *Vending Machine* (kiri) dan Tombol Power Notebook PC (kanan)

- 2) Buka aplikasi *Emoney_database.exe* (gambar 4.12) di, lalu hubungkan dengan cara klik *connect* pada menu di aplikasi (gambar 4.13)

Name	Date modified	Type	Size
Emoney_Database.exe	7/23/2017 9:19 PM	Application	238 KB
icudt51.dll	4/22/2013 11:03 PM	DLL File	21,854 KB
icuin51.dll	4/22/2013 11:03 PM	DLL File	3,291 KB
icuuc51.dll	4/22/2013 11:03 PM	DLL File	1,933 KB
libgcc_s_dw2-1.dll	4/18/2013 1:18 AM	DLL File	533 KB
libstdc++-6.dll	4/18/2013 1:19 AM	DLL File	967 KB
libwinpthread-1.dll	4/18/2013 12:26 AM	DLL File	73 KB
Qt5Core.dll	5/27/2017 3:01 PM	DLL File	4,500 KB
Qt5Gui.dll	2/2/2014 4:38 AM	DLL File	4,517 KB
Qt5SerialPort.dll	2/2/2014 5:06 AM	DLL File	114 KB
Qt5Sql.dll	2/2/2014 4:32 AM	DLL File	257 KB
Qt5Widgets.dll	2/2/2014 4:45 AM	DLL File	6,136 KB
settings.ini	8/21/2017 4:03 AM	Configuration sett...	1 KB

Gambar 4.12 Aplikasi *Emoney_database.exe*



Gambar 4.13 Hubungkan Aplikasi dengan *Vending Machine*

- 3) Pembeli datang ke administrator untuk membeli kartu *E-Money*. Perangkat yang digunakan administrator adalah PC, aplikasi *Emoney_database.exe*, dan perangkat pembaca kartu *E-Money* (*RFID Reader*).
- 4) Administrator mendaftarkan kartu dengan menempelkan kartu *E-Money* ke *RFID reader*, setelah kartu teridentifikasi lalu administrator mengisi saldo kartu tersebut melalui aplikasi *Emoney_database.exe*, setelah selesai maka kartu sudah dapat digunakan sebagai kartu *E-Money* karena sudah terdaftar dalam database sehingga kartu sudah dapat diberikan ke pembeli untuk dia gunakan saat melakukan transaksi di unit *vending machine*.
- 5) Pembeli datang ke *vending machine* untuk membeli minuman
- 6) Apabila kartu *E-Money* ditempelkan pada *E-Money reader* (*RFID reader*) *vending machine* maka *text LCD vending machine* akan menampilkan isi saldo kartu tersebut.
- 7) Apabila tombol pilihan minuman ditekan, maka:
 - a) *Text LCD 16x2* pada *vending machine* akan menampilkan “Sorry SOLD OUT” jika persediaan minuman yang dipilih tidak tersedia.

- b) Jika persediaan minuman yang dipilih tersedia, maka *text* LCD akan menampilkan “Please...TAP YOUR CARD”.
- 8) Setelah kartu ditempelkan, maka *box* tempat pengambilan minuman akan bergerak menuju koordinat minuman kaleng yang dipilih, kemudian *conveyor* pada *storage* minuman kaleng akan bergerak sehingga minuman kaleng tersebut berpindah ke dalam *box* tempat pengambilan minuman. Setelah *box* tempat pengambilan minuman terisi minuman kaleng, maka *box* tempat pengambilan minuman akan bergerak menuju tempat pengambilan minuman dan pintu akan terbuka sehingga pembeli dapat mengambil minuman tersebut. Setelah minuman diambil oleh pembeli, maka pintu akan tertutup dan *text* LCD 16x2 akan menampilkan “Thank You” lalu sisa saldo juga ditampilkan.

4.2. Analisis Data Penelitian

Saat proses pengujian telah didapat beberapa data yang dibutuhkan untuk diolah sebagai proses dalam sistem. Proses pengambilan data dilakukan dengan cara pengamatan simulasi dan hasilnya dicatat.

4.2.1. Pengujian *Actuator*

Pengujian *actuator* sumbu X dan Y dibutuhkan untuk mendapatkan nilai pulsa step yang tepat untuk menemukan koordinat minuman kaleng, selain itu juga untuk mengukur waktu tempuh maksimum yang paling cepat yang disesuaikan dengan kemampuan motor stepper.

1) *Actuator sumbu-X:*

Pengujian bertujuan untuk mengetahui kecepatan maksimum yang bisa digunakan untuk *actuator* sumbu-X dan mencari nilai pulsa step yang dibutuhkan untuk mencapai lokasi koordinat tertentu.

a) Jenis actuator: Motor Stepper NEMA17. Spesifikasi:

- (1) *1,8 degree step angle*
- (2) 200 step per-rotasi
- (3) *Holding torque = 0.6 Nm*
- (4) Tegangan masukan = 12 – 36 V
- (5) Arus = 1,96 A

b) Driver A4988

c) Mode yang digunakan fullstep

d) 1 rotasi = 3.85cm

e) Panjang sumbu-X = 54 cm

Hasil Pengujian:

a) Pulsa yang dibutuhkan untuk memenuhi panjang sumbu-X = **2721 pulsa**

b) Actuator sumbu-X berfungsi normal jika lebar pulsa yang diberikan ≥ 416 mikrodetik, jika lebar pulsa < 416 mikro detik maka actuator akan kehilangan kekuatan untuk menggeser benda

c) Waktu tempuh titik kanan-kiri atau kiri-kanan :

Uji coba yang dilakukan mengambil sampel 10 kali dengan menghitung waktu tempuh pergerakan actuator yang bergerak dari kiri ke kanan, hasilnya terlihat pada tabel 4.2:

Tabel 4.2 Pengujian Waktu Tempuh Sumbu X

Pengujian ke-	Durasi (detik)
1	2.36
2	2.23
3	2.47
4	2.25
5	2.25
6	2.03
7	2.31
8	2.15
9	2.22
10	2.25

Rata-rata T = 2.25 detik

2) *Actuator sumbu-Y:*

Pengujian bertujuan untuk mengetahui kecepatan maksimum yang bisa digunakan untuk *actuator* sumbu-Y dan mencari nilai pulsa step yang dibutuhkan untuk mencapai lokasi koordinat tertentu.

a) Jenis actuator: Motor Stepper NEMA17. Spesifikasi:

- (1) 1,8 *degree step angle*
- (2) 200 step per-rotasi
- (3) *Holding torque* = Nm
- (4) Tegangan masukan = 12 – 36 V
- (5) Arus = 1,96 A

b) Driver A4988

c) Mode yang digunakan fullstep

d) 1 rotasi = 0.8cm

e) Panjang sumbu-Y = 54 cm

Hasil Pengujian :

- a) Pulsa yang dibutuhkan untuk memenuhi panjang sumbu-Y = **15000 pulsa**
- b) Actuator sumbu-Y berfungsi normal jika lebar pulsa yang diberikan ≥ 600 mikrodetik, jika lebar pulsa < 600 mikro detik maka actuator akan kehilangan kekuatan untuk menggeser benda
- c) Waktu tempuh titik bawah-atas atau atas-bawah

Uji coba yang dilakukan mengambil sampel 10 kali dengan menghitung waktu tempuh pergerakan actuator yang bergerak dari bawah ke atas, hasilnya terlihat pada table 4.3:

Tabel 4.3 Pengujian Waktu Tempuh Sumbu Y

Pengujian ke-	Dur (detik)
1	18.73
2	18.76
3	18.37
4	18.61
5	18.40
6	18.38
7	18.43
8	18.40
9	18.65
10	18.39

Rata-rata T = 18.51

Koordinat minuman kaleng berdasarkan step pergerakan motor stepper dapat dilihat pada tabel 4.4.

Tabel 4.4 Koordinat minuman kaleng berdasarkan step pergerakan motor stepper

Nomor Minuman Kaleng	X (step)	Y (step)
1	1510	8400
2	2108	8400
3	2721	8400
4	1510	4250
5	2108	4250
6	2721	4250
7	1510	0
8	2108	0
9	2721	0

3) *Actuator* sumbu-Z:

Pengujian bertujuan untuk mencoba fungsi kerja motor DC yang berperan sebagai penggerak *actuator* sumbu Z dengan mencari nilai tegangan minimum agar motor dapat beroperasi. Ujicoba dilakukan dengan memberikan daya terhadap motor agar dapat bergerak lalu diukur dan dicatat tegangannya. Hasil pengukuran menunjukkan *actuator* sumbu Z dapat beroperasi sesuai tabel 4.5.

Tabel 4.5 Hasil Pengukuran Tegangan *Actuator* Sumbu Z

Percobaan Ke-	Tegangan	Aksi Motor DC
1	1 Volt	Tidak bergerak
2	3 Volt	Tidak bergerak
3	5 Volt	Motor berputar tapi tidak mampu menggerakkan beban <i>box</i>
4	7 Volt	Motor berputar tapi tidak mampu menggerakkan beban <i>box</i>
5	9 Volt	Motor bergerak, mampu menggerakkan beban <i>box</i>

Dari hasil percobaan ke-5, telah didapat nilai tegangan masukkan yang dibutuhkan motor DC agar dapat bergerak sesuai fungsi *actuator* sumbu Z adalah

9 Volt.

4) *Actuator* Pintu Tempat Pengambilan Minuman:

Pengujian bertujuan untuk mencoba fungsi kerja motor DC yang berperan sebagai penggerak *actuator* pintu tempat pengambilan minuman kaleng dengan mencari nilai tegangan minimum agar motor dapat beroperasi. Ujicoba dilakukan dengan memberikan daya terhadap motor agar dapat bergerak lalu diukur dan dicatat tegangannya. Hasil pengukuran ditunjukkan pada tabel 4.6.

Tabel 4.6 Hasil Pengukuran Tegangan *Actuator* Pintu Tempat Pengambilan Minuman Kaleng

Percobaan Ke-	Tegangan	Aksi Motor DC
1	1 Volt	Tidak bergerak
2	3 Volt	Tidak bergerak
3	5 Volt	Motor berputar, tapi tidak mampu menggerakkan mekanik pintu
4	7 Volt	Motor berputar, tapi tidak mampu menggerakkan mekanik pintu
5	9 Volt	Motor berputar, dan mampu menggerakkan mekanik pintu

Dari hasil percobaan ke-5, telah didapat nilai tegangan masukkan yang dibutuhkan motor DC agar dapat bergerak sesuai fungsi *actuator* pintu tempat pengambilan minuman adalah **9 Volt**.

4.2.2. Pengujian Sensor Cahaya

Ujicoba sensor cahaya dilakukan untuk mencoba fungsi rangkaian sensor cahaya, fotodiode, dan LED infra merah sebagai pendeteksi keberadaan minuman kaleng. Sebelum melakukan percobaan, sensor telah dipasang sesuai letaknya di dalam *vending machine*. Jarak LED Infra merah ke Fotodiode adalah 6,3 cm. Saat cahaya yang masuk ke fotodiode terhalang dan tidak terhalang oleh minuman kaleng harus memenuhi logika digital 1 dan 0. Tegangan Fotodiode diukur saat

menerima dan tidak menerima cahaya dari LED infra merah, dan hasilnya sesuai tabel 4.7. Hasil pengukuran keluaran rangkaian sensor terlihat pada tabel 4.8

Tabel 4.7 Pengukuran Tegangan Fotodioda

Kondisi	Tegangan Pada Fotodioda
Fotodioda menerima cahaya LED infra merah	2,409 V
Fotodioda tidak menerima cahaya LED infra merah (terhalang minuman kaleng)	4,570 V

Tabel 4.8 Pengukuran Tegangan Keluaran Rangkaian Sensor Cahaya

Kondisi	Tegangan Keluaran Rangkaian Sensor	Logika Digital
Fotodioda Terhalang Minuman Kaleng	1,292 V	0
Fotodioda Tidak Terhalang Minuman Kaleng	3,481 V	1

Hasil tegangan keluaran rangkaian sensor cahaya jika dibaca pada arduino sudah memenuhi logika digital 1 dan 0, maka rangkaian sensor sudah dapat digunakan untuk mendeteksi keberadaan minuman kaleng.

4.2.3. Pengujian *E-Money*

E-Money diujicobakan fungsinya dimulai dari pembacaan kartu untuk registrasi, pengisian saldo, sampai dengan simulasi transaksi pembelian.

- 1) Pengujian pembacaan kartu *E-Money* dilakukan untuk memastikan kartu bisa diidentifikasi oleh sistem *E-Money* agar terdaftar sebagai kartu yang dapat digunakan. Pengujian dilakukan dengan menempelkan kartu pada pembaca kartu *E-Money*, lalu hasilnya terlihat pada tampilan aplikasi administrator.
- 2) Pengujian pengisian saldo dilakukan melalui aplikasi administrator.

- 3) Pengujian pembelian minuman kaleng diujicobakan dengan simulasi pembelian pada unit *vending machine*.

4.2.4. Waktu Pelayanan *Vending machine* Minuman Kaleng

Kemampuan pelayanan penyediaan minuman kaleng oleh *vending machine* diukur dengan menghitung waktu setiap proses pelayanannya. Hasilnya terlihat pada tabel 4.9.

Tabel 4.9 Hasil Pengujian Waktu Pelayanan Pengambilan Minuman Kaleng

No.	Nomor Minuman	Durasi (detik)
1.	Minuman di nomor 1	37.90
2.	Minuman di nomor 2	38.50
3.	Minuman di nomor 3	41.10
4.	Minuman di nomor 4	20.75
5.	Minuman di nomor 5	20.90
6.	Minuman di nomor 6	21.22
7.	Minuman di nomor 7	10.10
8.	Minuman di nomor 8	10.68
9.	Minuman di nomor 9	11.20

4.3. Pembahasan

Hasil dari semua ujicoba secara keseluruhan alat berfungsi sesuai rancangan. Mekanik dapat menunjang pergerakan *Box* pengambil minuman, Elektronik berfungsi dengan benar, Sistem *E-Money* dapat berfungsi sebagai alat transaksi pembayaran.

4.4. Aplikasi Hasil Penelitian

Vending machine ini dapat diaplikasikan sebagai mesin penjual minuman kaleng yang di tempatkan pada kantin sekolah, kantor, arena permainan dimana kasir bertindak sebagai administrator *E-Money*.

BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Setelah melihat hasil dari uji coba dan membandingkan dengan rumusan masalahnya maka kesimpulan dari penelitian ini adalah:

- 1) Prototipe *vending machine* minuman kaleng berbasis Arduino Mega 2560 dapat berfungsi sesuai rancangan untuk mengambil/menyediakan minuman kaleng kepada pembeli.
- 2) Sistem *E-Money* yang telah dibuat dapat berfungsi untuk alat bayar saat transaksi pembelian minuman kaleng pada unit *vending machine*.

5.2. Saran

Selama proses penelitian, peneliti telah memikirkan beberapa hal yang harusnya dapat dimaksimalkan dan ide baru yang dapat diterapkan agar *vending machine* dapat lebih baik. Peneliti menyarankan untuk penelitian lanjutan supaya mempercepat pergerakan *actuator* agar semakin memaksimalkan dalam hal kecepatan pelayanan. Kemudian peneliti menyarankan komunikasi data yang terjadi antara server database dengan unit *vending machine* dapat terjalin melalui internet dan satu server dapat melayani lebih dari satu unit *vending machine*.

DAFTAR PUSTAKA

- Fakultas Teknik. 2015. *Buku Panduan Penyusunan Skripsi dan Non Skripsi*. Jakarta: Fakultas Teknik, Universitas Negeri Jakarta.
- Finkenzeller, Klaus. 2010. *RFID Handbook Fundamentals and Applications in Contactless Smart Cards, Radio Frequency Identification and Near-Field Communication*. Ed ke-3. Chichester: John Wiley & Sons, Ltd.
- Hughes, 2006. Austin. *Electric Motors and Drives Fundamentals, Types and Applications*. Ed ke-3. Oxford: Elsevier Ltd.
- Margolis, Michael. 2011. *Arduino Cookbook*. Sebastopol: O'Reilly Media, Inc.
- Segrave, Kerry. 2002. *Vending machines: An American Social History*. Jefferson: McFarland & Company, Inc.
- Wildi, Theodore. 2002. *Electrical Machines, Drives, and Power Systems*. Ed ke-5. Upper Saddle River: Pearson Education, Inc.
- <http://www.japan-guide.com/e/e2010.html> diakses pada 14 Juli 2017
- <https://www.arduino.cc/> diakses pada 14 Juli 2017
- <https://www.mysql.com/products/workbench/> diakses pada 14 Juli 2017
- <http://kbbi.web.id/> diakses pada 14 Juli 2017
- <https://www.paysafecard.com/en/corporate/blog/detail/what-is-E-Money/> diakses pada 27 Juli 2017
- <http://sis.binus.ac.id/2014/04/12/radio-frequency-identification-rfid/> diakses pada 27 Juli 2017
- http://lexicon.ft.com/Term?term=e_money diakses pada 27 Juli 2017
- <https://www.electrical4u.com/dc-motor-or-direct-current-motor/> diakses pada 27 Juli 2017
- <http://www.belajarduino.com/2016/06/how-to-connect-1602-2004-iic-i2c-led-to.html> diakses pada 27 Juli 2017
- http://eprints.undip.ac.id/4886/1/Sensor_dan_Transduser.pdf diakses pada 27 Juli 2017
- <http://elektronika-dasar.web.id/limit-switch-dan-saklar-push-on/> diakses pada 27 Juli 2017
- <http://elektronikadasar.info/sensor-cahaya.htm> diakses pada 27 Juli 2017

<http://howtomechatronics.com/tutorials/arduino/how-to-control-stepper-motor-with-a4988-driver-and-arduino/> diakses pada 27 Juli 2017

<http://www.stronglink-rfid.com/en/rfid-readers/sl500.html> diakses pada 21 Agustus 2017

<https://www.mifare.net/> diakses pada 21 Agustus 2017

LAMPIRAN

Lampiran 1. Dokumentasi

Lampiran 2. Gambar Teknik

Lampiran 3 Data-Data Pengukuran

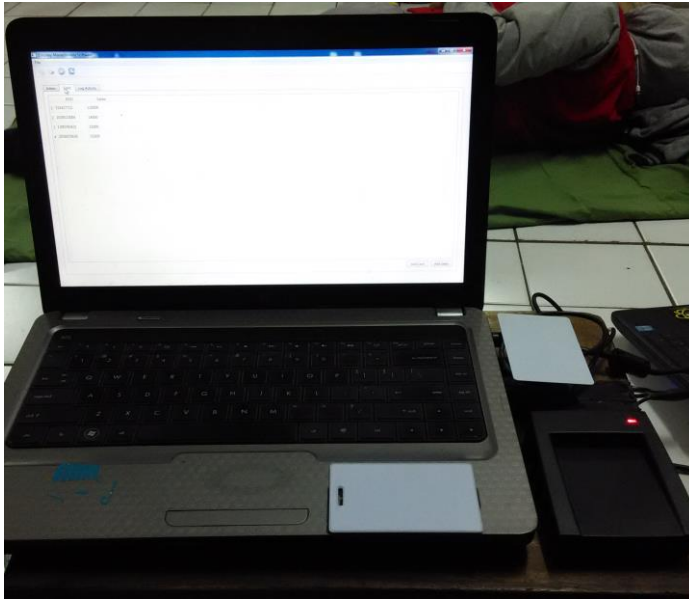
Lampiran 4 *Source Code*

Lampiran 1. Dokumentasi

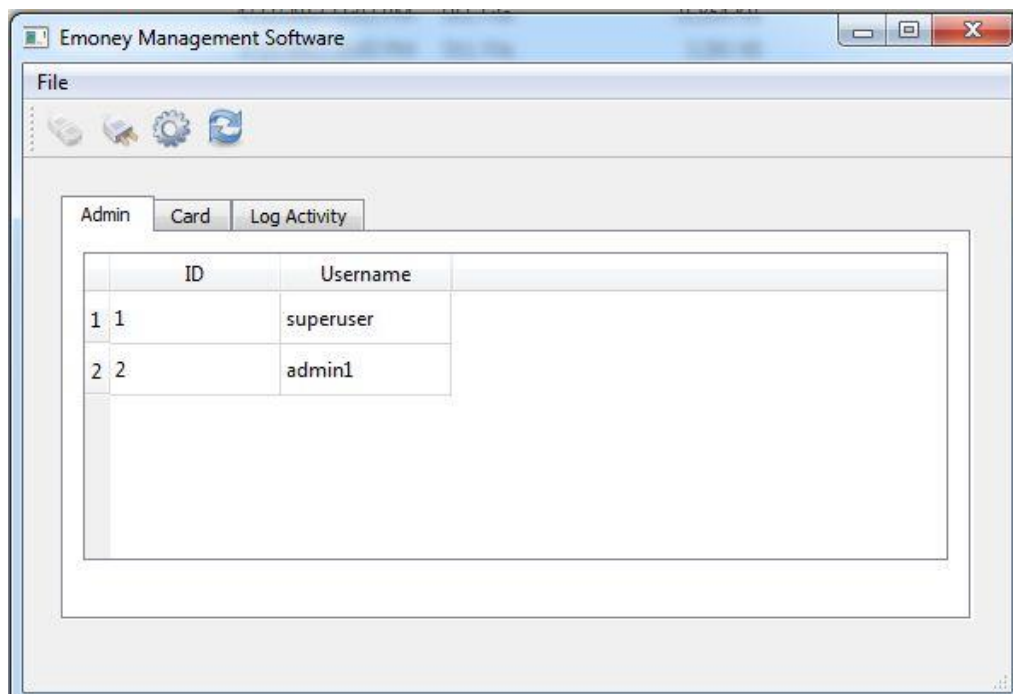
Hasil pembuatan alat:

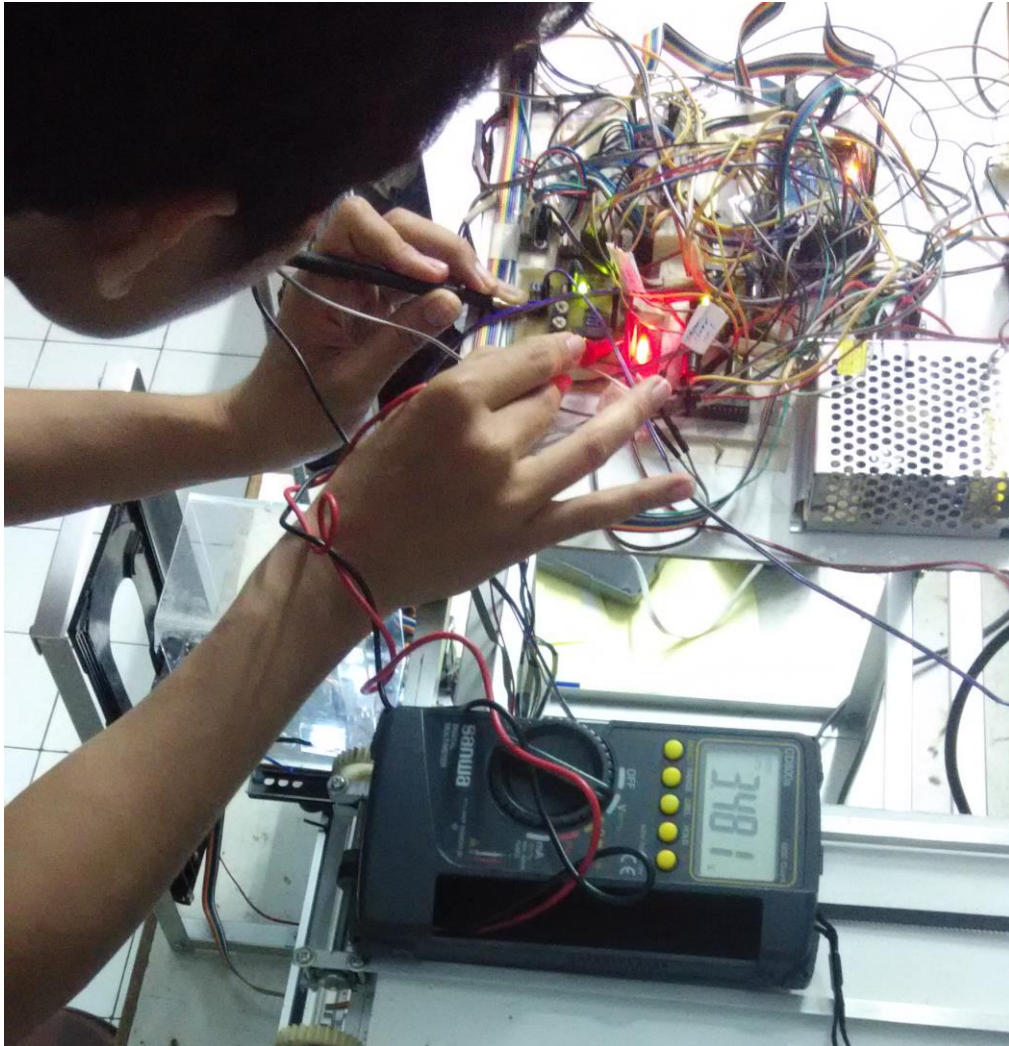


Perangkat yang digunakan administrator *E-Money*(PC, aplikasi *Emoney_Database.exe*, *RFID reader plug and play*:



Aplikasi *Emoney_Database.exe* yang terdapat pada PC:



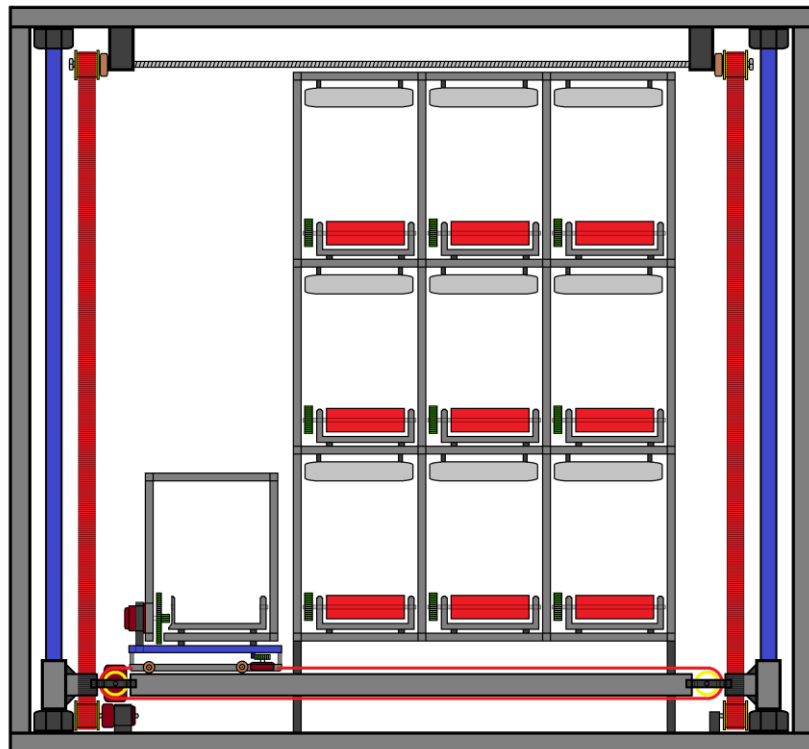
Dokumentasi Saat Pengukuran Rangkaian Elektronik:

Lampiran 2. Gambar Teknik

Desain Mekanik *Vending machine*:

Keterangan:

	Pulley
	Gear
	Timing Belt / Conveyor Belt
	Linear Gear / Linear Sliding
	Motor DC Geared
	Bearing
	Frame
	Penyambung



Lampiran 3. Data-Data Pengukuran

actuator sumbu-X:

- a) Driver A4988
- b) Mode yang digunakan fullstep
- c) 1 rotasi = 3.85cm
- d) Panjang sumbu-X = 54 cm
- e) Pulsa yang dibutuhkan untuk memenuhi panjang sumbu-X = **2721 pulsa**
- f) Actuator sumbu-X berfungsi normal jika lebar pulsa yang diberikan ≥ 416 mikrodetik, jika lebar pulsa < 416 mikro detik maka actuator akan kehilangan kekuatan untuk menggeser benda
- g) Data pengujian Waktu Tempuh Sumbu X

Pengujian ke-	Durasi (detik)
1	2.36
2	2.23
3	2.47
4	2.25
5	2.25
6	2.03
7	2.31
8	2.15
9	2.22
10	2.25

Rata-rata T = 2.25 detik

actuator sumbu-Y:

- a) Driver A4988
- b) Mode yang digunakan fullstep
- c) 1 rotasi = 0.8cm
- d) Panjang sumbu-Y = 54 cm

- e) Pulsa yang dibutuhkan untuk memenuhi panjang sumbu-Y = **15000 pulsa**
- f) Actuator sumbu-Y berfungsi normal jika lebar pulsa yang diberikan ≥ 600 mikrodetik, jika lebar pulsa < 600 mikro detik maka actuator akan kehilangan kekuatan untuk menggeser benda
- g) Data Pengujian Waktu Tempuh Sumbu Y:

Pengujian ke-	Dur (detik)
1	18.73
2	18.76
3	18.37
4	18.61
5	18.40
6	18.38
7	18.43
8	18.40
9	18.65
10	18.39

Rata-rata T = 18.51

Data koordinat minuman kaleng berdasarkan step pergerakan motor stepper:

Nomor Minuman Kaleng	X (step)	Y (step)
1	1510	8400
2	2108	8400
3	2721	8400
4	1510	4250
5	2108	4250
6	2721	4250
7	1510	0
8	2108	0
9	2721	0

actuator sumbu-Z:**Hasil Pengukuran Tegangan Actuator Sumbu Z:**

Percobaan Ke-	Tegangan	Aksi Motor DC
1	1 Volt	Tidak bergerak
2	3 Volt	Tidak bergerak
3	5 Volt	Motor berputar tapi tidak mampu menggerakkan beban <i>box</i>
4	7 Volt	Motor berputar tapi tidak mampu menggerakkan beban <i>box</i>
5	9 Volt	Motor bergerak, mampu menggerakkan beban <i>box</i>

Dari hasil percobaan ke-5, telah didapat nilai tegangan masukan yang dibutuhkan motor DC agar dapat bergerak sesuai fungsi *actuator* sumbu Z adalah

9 Volt.

Actuator Pintu Tempat Pengambilan Minuman:**Hasil Pengukuran Tegangan Actuator Pintu Tempat Pengambilan Minuman Kaleng:**

Percobaan Ke-	Tegangan	Aksi Motor DC
1	1 Volt	Tidak bergerak
2	3 Volt	Tidak bergerak
3	5 Volt	Motor berputar, tapi tidak mampu menggerakkan mekanik pintu
4	7 Volt	Motor berputar, tapi tidak mampu menggerakkan mekanik pintu
5	9 Volt	Motor berputar, dan mampu menggerakkan mekanik pintu

Dari hasil percobaan ke-5, telah didapat nilai tegangan masukan yang dibutuhkan motor DC agar dapat bergerak sesuai fungsi *actuator* pintu tempat pengambilan minuman adalah **9 Volt.**

Pengujian Sensor Cahaya

Pengukuran Tegangan Fotodiode:

Kondisi	Tegangan Pada Fotodiode
Fotodiode menerima cahaya LED infra merah	2,409 V
Fotodiode tidak menerima cahaya LED infra merah (terhalang minuman kaleng)	4,570 V

Pengukuran Tegangan Keluaran Rangkaian Sensor Cahaya:

Kondisi	Tegangan Keluaran Rangkaian Sensor	Logika Digital
Fotodiode Terhalang Minuman Kaleng	1,292 V	0
Fotodiode Tidak Terhalang Minuman Kaleng	3,481 V	1

Lampiran 4. Source Code

1. Source Code pada *Sketch* Arduino IDE:

```

#include <Wire.h>
#include <LiquidCrystal_I2C.h>

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

#define SS_PIN 53
#define RST_PIN 5

LiquidCrystal_I2C lcd(0x3E, 16, 2);

MFRC522 rfid(SS_PIN, RST_PIN); // Instance of the
class

MFRC522::MIFARE_Key key;

// Init array that will store new NUID
byte nuidPICC[3];

// Tombol Pilihan Minuman
const int pb_btn[9] = {
    28, 29, 30,
    25, 26, 27,
    22, 23, 24
};

const int harga[9] = {
    5000, 5000, 5000,
    8000, 8000, 8000,
    10000, 10000, 10000
};

// Limit Switch Minuman
const int sensor[9] = {
    31, 32, 33,
    34, 35, 36,
    37, 38, 39
};

int dir_x, dir_y, price;
////////////////////////////////////
/** TOMBOL OK & CANCEL **/
////////////////////////////////////
const int btn_ok = 40, btn_cncl = 41;
bool b_btn = false;
int cnt_tap;

```

```

char DataSerial1[50], DataSerial2[50],
DataSerial3[50];
char DataSerial[50], *temp;

bool debug = false;

const int step_motor1 = 6; ///  

const int dir_motor1 = 7;  

const int step_motor2 = 8; ///  

const int dir_motor2 = 9;

//kalibrasi step jarak  

const int x[4] = {0, 1510, 2108, 2721};  

const int y[4] = {0, 8400, 4250, 0};

const int motor_z_maju = A4;  

const int motor_z_mundur = A5;  

const int lm_z_maju = A6;  

const int lm_z_mundur = A7;  

const int motor_conveyor = A8;  

const int sc_box = A9; //  

Cahaya

const int pintu_buka = A10;  

const int pintu_tutup = A11;  

const int lm_buka = A12;  

const int lm_tutup = A13;

//const int lm_kanan = 42;  

const int lm_kiri = 43;  

//const int lm_atas = 44;  

const int lm_bawah = 45;

const int buzz = 46;

unsigned long data_card;

int glb_temp = 0;

void setup() {
  Serial.begin(9600);
  lcd.begin();
  lcd.backlight();
  SPI.begin(); // Init SPI bus
  rfid.PCD_Init(); // Init MFRC522

  for (int i = 0; i < 9; i++) {
    pinMode(pb_btn[i], INPUT);
    pinMode(sensor[i], INPUT);
  }
}

```

```

}

pinMode(btn_ok, INPUT);
pinMode(btn_cncl, INPUT);

pinMode(step_motor1, OUTPUT);
pinMode(dir_motor1, OUTPUT);
pinMode(step_motor2, OUTPUT);
pinMode(dir_motor2, OUTPUT);

pinMode(motor_z_maju, OUTPUT);
pinMode(motor_z_mundur, OUTPUT);
pinMode(motor_conveyor, OUTPUT);
pinMode(sc_box, INPUT);

pinMode(pintu_buka, OUTPUT);
pinMode(lm_buka, INPUT);
pinMode(pintu_tutup, OUTPUT);
pinMode(lm_tutup, INPUT);
// pinMode(lm_minuman, INPUT);

// pinMode(lm_kanan, INPUT);
pinMode(lm_kiri, INPUT);
// pinMode(lm_atas, INPUT);
pinMode(lm_bawah, INPUT);

pinMode(buzz, OUTPUT);

// while(1) {
//   int val2 = digitalRead(lm_tutup);
//   Serial.print("LM Tutup : ");
Serial.println(val2);
// }

digitalWrite(dir_motor1, LOW);
digitalWrite(dir_motor2, LOW);
digitalWrite(step_motor1, LOW);
digitalWrite(step_motor2, LOW);
}

void loop() {
  readSerial();      // Proses baca data dari
program desktop
  Read_RFID();      // Proses kirim data ke
program desktop
  ProcessButton();
  awal();
}

```

```

void ProcessButton() {
    int val_btn;
    val_btn = digitalRead(sc_box);
    if (val_btn == 1) {
        digitalWrite(buzz, LOW);
        if (!b_btn) {
            for (int i = 0; i < 9; i++) {
                val_btn = digitalRead(pb_btn[i]);
                if (val_btn) {
                    dir_y = i/3;
                    if ( i < 3) dir_x = i;
                    else if (i >= 3  && i < 6) dir_x = i-3;
                    else if (i >= 6  && i < 9) dir_x = i-6;
                    dir_x++;
                    dir_y++;
                    price = harga[i];
                    val_btn = digitalRead(sensor[i]);
                }
                // val_btn = 1;
                if (!val_btn) {
                    // beri tanda seru (!) didepan val_btn untuk aktif
                    low

                    lcd.clear();
                    for (int i = 500; i >= 0; i--=1) {
                        lcd.setCursor(0,0);
                        lcd.print("ARE YOU SURE ?");
                        if (debug) {
                            lcd.setCursor(15,0);
                            lcd.print(i);
                            Serial.println(i);
                            lcd.setCursor(0,1);
                            lcd.print("X = ");
                            lcd.setCursor(4, 1);
                            lcd.print(dir_x);
                            lcd.setCursor(5,1);
                            lcd.print(",Y = ");
                            lcd.setCursor(10, 1);
                            lcd.print(dir_y);
                        } else {
                            lcd.setCursor(0,1);
                            lcd.print("NO          YES");
                        }
                    }

                    val_btn = digitalRead(btn_ok);
                    if (val_btn) {
                        b_btn = true;
                        cnt_tap = 1000;
                        lcd.clear();
                        break;
                    }
                }
            }
        }
    }
}

```

```

        val_btn = digitalRead(btn_cncl);
        if (val_btn) {
            b_btn = false;
            break;
        }
    }
    lcd.clear();
} else {
    if (debug) {
        Serial.println("Sorry Habis");
    }
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Sorry ...");
    lcd.setCursor(0,1);
    lcd.print("SOLD OUT !!");
    delay(5000);
    lcd.clear();
}
}
if (debug) {
    Serial.println();
}
}
}
} else if (val_btn == 0) {
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Please Clear ,,");
    lcd.setCursor(0,1);
    lcd.print("The Box ...");
    delay(5000);
    lcd.clear();
    digitalWrite(buzz, HIGH);
}
}

void awal() {
    if (b_btn) {
        if (cnt_tap > 0) {
            lcd.setCursor(0,0);
            lcd.print("Please ...");
            lcd.setCursor(0,1);
            lcd.print("TAP YOUR CARD ...");
            cnt_tap--;
            if (cnt_tap == 0) {
                lcd.clear();
                lcd.setCursor(0,0);
                lcd.print("Sorry ...");
            }
        }
    }
}

```

```

        lcd.setCursor(0,1);
        lcd.print("Try again ...");
        delay(5000);
        lcd.clear();
        b_btn = false;
    }
} else {
    b_btn = false;
}
} else {
//    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Welcome to ...");
    lcd.setCursor(7,1);
    lcd.print("Emoney VM");

    int val = 0;
    int val1 = 0; bool bVal1 = true;
    int val2 = 0; bool bVal2 = true;
    while(val == 0) {
        val = digitalRead(lm_z_mundur);
        digitalWrite(motor_z_maju, LOW);
        digitalWrite(motor_z_mundur, HIGH);
    }
    digitalWrite(motor_z_maju, LOW);
    digitalWrite(motor_z_mundur, LOW);

    val = digitalRead(sc_box);
    if (val == 0) {
        glb_temp = 0;
        val1 = digitalRead(lm_buka);
        if (val1 == 0) {
            digitalWrite(pintu_buka, HIGH);
            digitalWrite(pintu_tutup, LOW);
        } else if (val1 == 1) {
            digitalWrite(pintu_buka, LOW);
            digitalWrite(pintu_tutup, LOW);
        }
    }
    } else if (val == 1) {
        val2 = digitalRead(lm_tutup);
        if (val2 == 0) {
            if (glb_temp == 0) {
                digitalWrite(pintu_buka, LOW);
                digitalWrite(pintu_tutup, HIGH);
            } else {
                glb_temp--;
                digitalWrite(pintu_buka, LOW);
                digitalWrite(pintu_tutup, LOW);
            }
        }
    }
}

```



```

    } else if (val2 == 1) {
        digitalWrite(pintu_buka, LOW);
        digitalWrite(pintu_tutup, LOW);
    }
}

val1 = 0; bVal1 = true;
val2 = 0; bVal2 = true;
digitalWrite(dir_motor1, HIGH);
digitalWrite(dir_motor2, LOW);
while(val1 == 0 || val2 == 0) {
    val1 = digitalRead(lm_kiri);
    val2 = digitalRead(lm_bawah);
    if (!val1) digitalWrite(step_motor1, HIGH);
    if (!val2) digitalWrite(step_motor2, HIGH);
    delayMicroseconds(600);
    if (!val1) digitalWrite(step_motor1, LOW);
    if (!val2) digitalWrite(step_motor2, LOW);
    delayMicroseconds(600);
}
}
}

void Read_RFID() {
    // Look for new cards
    if ( ! rfid.PICC_IsNewCardPresent() )
        return;

    // Verify if the NUID has been readed
    if ( ! rfid.PICC_ReadCardSerial() )
        return;

    // Serial.print(F("PICC type: "));
    MFRC522::PICC_Type piccType =
    rfid.PICC_GetType(rfid.uid.sak);
    //
    Serial.println(rfid.PICC_GetTypeName(piccType));

    // Check is the PICC of Classic MIFARE type
    if (piccType != MFRC522::PICC_TYPE_MIFARE_MINI
    &&
        piccType != MFRC522::PICC_TYPE_MIFARE_1K &&
        piccType != MFRC522::PICC_TYPE_MIFARE_4K) {
    // Serial.println(F("Your tag is not of type
    MIFARE Classic."));
        return;
    }

    for (byte i = 0; i < 4; i++) {

```

```

    nuidPICC[i] = rfid.uid.uidByte[i];
}
Print_RFID_Dec(rfid.uid.uidByte, rfid.uid.size);
Serial.println();

rfid.PICC_HaltA();
rfid.PCD_StopCrypto1();
}

void Print_RFID_Dec(byte *buffer, byte bufferSize)
{
    data_card = 0;
    data_card += (long)buffer[3]<<24;
    data_card += (long)buffer[2]<<16;
    data_card += (long)buffer[1]<<8;
    data_card += (long)buffer[0];

    data_card = *((long*)buffer);

    if (b_btn) {
        Serial.print("*SET;");
        Serial.print(data_card, DEC);
        Serial.print(";");
        Serial.print(price);
        Serial.println("#");
    } else {
        Serial.print("*GET;");
        Serial.print(data_card, DEC);
        Serial.println("#");
    }
}

void readSerial() {
    while (Serial.available()) {
        char c[2]; c[1] = '\0'; c[0] = Serial.read();
        if (c[0] == 0x0d || c[0] == 0x0a) continue;

        strcat(DataSerial, c);

        if (strchr(DataSerial, '#') != NULL) {
            temp = strtok (DataSerial, ";#");
            int cnt = 0;
            memset(DataSerial1, '\0', 50);
            memset(DataSerial2, '\0', 50);
            memset(DataSerial3, '\0', 50);
            while (temp != NULL) {

                if (cnt == 0) strcpy(DataSerial1, temp);
                if (cnt == 1) strcpy(DataSerial2, temp);
            }
        }
    }
}

```

```

        if (cnt == 2) strcpy(DataSerial3, temp);
        temp = strtok (NULL, ";#");
        cnt++;
    }

//      if (strcmp(DataSerial1, "enroll") == 0 &&
strcmp(DataSerial2, "0") != 0) {erl = true; fng =
false;}
//      else if (strcmp(DataSerial1, "finger") ==
0 && strcmp(DataSerial2, "0") == 0) {erl = false;
fng = true;}

        if (DataSerial1[0] == 'G' && DataSerial1[1]
== 'E' && DataSerial1[2] == 'T') {
            lcd.clear();
            lcd.setCursor(0,0);
            lcd.print(DataSerial2);
            lcd.setCursor(0,1);
            lcd.print(DataSerial3);
            delay(5000);
            lcd.clear();
        } else if (DataSerial1[0] == 'S' &&
DataSerial1[1] == 'E' && DataSerial1[2] == 'T') {
            if (DataSerial3[0] == '1') {
                lcd.clear();
                lcd.setCursor(0,0);
                lcd.print("Processing ...");
                lcd.setCursor(0,1);
                lcd.print("Wait a minute ...");

                stepper(0, x[dir_x], y[dir_y]);
                getMinuman();
                stepper(1, x[dir_x], y[dir_y]);
                kasihMinuman();

                lcd.clear();
                lcd.setCursor(0,0);
                lcd.print("Thank you !!");
                delay(5000);
                lcd.clear();

                Serial.print("*GET;");
                Serial.print(data_card,DEC);
                Serial.println("#");

                b_btn = false;
            } else if (DataSerial3[0] == '0') {
                lcd.clear();
            }
        }

```

```

        lcd.setCursor(0,0);
        lcd.print("Sorry ...");
        lcd.setCursor(0,1);
        lcd.print("PROCESS ERROR ...");
        delay(5000);
        lcd.clear();

        Serial.print("*GET;");
        Serial.print(data_card,DEC);
        Serial.println("#");

        b_btn = false;
    }
}

memset(DataSerial, '\0', 50);
}
}

void stepper(int dir, int spd_x, int spd_y) {
    digitalWrite(dir_motor1, dir);
    digitalWrite(dir_motor2, !dir);
    int cnt_x = 0, cnt_y = 0;
    if (spd_x > spd_y) {
        for (int i = 0; i < spd_x; i++) {
            if (cnt_x < spd_x) digitalWrite(step_motor1,
HIGH);
            if (cnt_y < spd_y) digitalWrite(step_motor2,
HIGH);
            delayMicroseconds(600);           //Kecepatan
putar Stepper
            if (cnt_x < spd_x) digitalWrite(step_motor1,
LOW);
            if (cnt_y < spd_y) digitalWrite(step_motor2,
LOW);
            delayMicroseconds(600);           //Kecepatan
putar Stepper
            cnt_x++; cnt_y++;
        }
    } else {
        for (int i = 0; i < spd_y; i++) {
            if (cnt_x < spd_x) digitalWrite(step_motor1,
HIGH);
            if (cnt_y < spd_y) digitalWrite(step_motor2,
HIGH);
            delayMicroseconds(600);           //Kecepatan
putar Stepper

```

```

        if (cnt_x < spd_x) digitalWrite(step_motor1,
LOW);
        if (cnt_y < spd_y) digitalWrite(step_motor2,
LOW);
        delayMicroseconds(600);          //Kecepatan
putar Stepper
        cnt_x++; cnt_y++;
    }
}
}

```

```

void getMinuman() {
    int val = 0;
    while(val == 0) {
        val = digitalRead(lm_z_maju);
        digitalWrite(motor_z_maju, HIGH);
        digitalWrite(motor_z_mundur, LOW);
    }
    digitalWrite(motor_z_maju, LOW);
    digitalWrite(motor_z_mundur, LOW);

    val = 0;
    while(val == 0) {
        val = digitalRead(sc_box);
        digitalWrite(motor_conveyor, HIGH);
    }
    delay(5000);
    digitalWrite(motor_conveyor, LOW);

    val = 0;
    while(val == 0) {
        val = digitalRead(lm_z_mundur);
        digitalWrite(motor_z_maju, LOW);
        digitalWrite(motor_z_mundur, HIGH);
    }
    digitalWrite(motor_z_maju, LOW);
    digitalWrite(motor_z_mundur, LOW);
}

```

```

void kasihMinuman() {
    int val = 0;
    while(val == 0) {
        val = digitalRead(lm_buka);
        digitalWrite(pintu_buka, HIGH);
        digitalWrite(pintu_tutup, LOW);
    }
    digitalWrite(pintu_buka, LOW);
    digitalWrite(pintu_tutup, LOW);
}

```

```
val = digitalRead(sc_box);
while (val == 1) {
    val = digitalRead(sc_box);
    delay(1);
}
delay(10000);

val = 0;
while(val == 0) {
    val = digitalRead(lm_tutup);
    digitalWrite(pintu_buka, LOW);
    digitalWrite(pintu_tutup, HIGH);
}
digitalWrite(pintu_buka, LOW);
digitalWrite(pintu_tutup, LOW);
}
```

2. Source Code pada QT Creator:

global.h :

```

#ifndef GLOBAL_H
#define GLOBAL_H

#include <QDebug>
#include <QMessageBox>

#include <QSerialPort>
#include <QSerialPortInfo>

#include <QFile>
#include <QSettings>
#include <QVariant>

#include <QSqlError>
#include <QSqlDatabase>
#include <QSqlQuery>

#include <QLineEdit>
#include <QStandardItemModel>

#include <QTimer>
#include <QDateTime>

#define MAX_ADMIN 10
#define MAX_CARD 10

struct serial {
    QString name;
    int c_name_id;

    qint32 baudRate;
    int c_baudRate_id;
    QString stringBaudRate;

    QSerialPort::DataBits dataBits;
    int c_dataBits_id;
    QString stringDataBits;

    QSerialPort::Parity parity;
    int c_parity_id;
    QString stringParity;

    QSerialPort::StopBits stopBits;
    int c_stopBits_id;
    QString stringStopBits;

    QSerialPort::FlowControl flowControl;
    int c_flowControl_id;
    QString stringFlowControl;
};

struct database {
    QString host;
    QString db_name;
};

```

```

    QString username;
    QString password;
};

struct setting {
    struct serial serialCommunication;
    struct database Database;
};

//struct admin {
//    int id_admin;
//    QString username;
//    QString password;

////    QDateTime create_date;
////    QString create_by;
////    QDateTime update_date;
////    QString update_by;
//};

struct opt {
    QStringList id, username;
    int jml_operator;
};

struct card {
    QStringList rfid, saldo;
    int jml_card;
};

struct log_history {
    QStringList action, user_id, card_id, date;
    int jml_log;
};

#endif // GLOBAL_H

```

init_mysql.h :

```

#ifndef INIT_MYSQL_H
#define INIT_MYSQL_H

#include "global.h"

class init_mysql
{
public:
    init_mysql();

    QString host;
    QString db_name;
    QString user_name;
    QString password;

    QSqlDatabase connect_db();
    void close(QSqlDatabase db);
};

#endif // INIT_MYSQL_H

```

Logindialog.h :

```
#ifndef LOGINDIALOG_H
#define LOGINDIALOG_H

#include <QDialog>

#include "init_mysql.h"
#include <QSqlDatabase>
#include <QSqlQuery>
#include "mainwindow.h"

namespace Ui {
class LoginDialog;
}

class LoginDialog : public QDialog
{
    Q_OBJECT

public:
    explicit LoginDialog(QWidget *parent = 0);
    ~LoginDialog();

private slots:
    void on_pushButton_clicked();

private:
    Ui::LoginDialog *ui;
    init_mysql mysql;
    QSqlDatabase db;
    void getAdmin();
    QStringList user;
    MainWindow *mWindow;
};

#endif // LOGINDIALOG_H
```

mainwindow.h :

```
#ifndef MAINWINDOW_H
#define MAINWINDOW_H

#include <QMainWindow>
#include "global.h"
#include "settingsdialog.h"
#include "init_mysql.h"

namespace Ui {
class MainWindow;
}
```

```

}

class MainWindow : public QMainWindow
{
    Q_OBJECT

public:
    explicit MainWindow(QWidget *parent = 0, int user_id = 0);
    ~MainWindow();

private slots:
    void readData_Serial(); // Baca
data dari arduino

    void on_actionSetting_triggered();

    void on_actionConnect_triggered();

    void on_actionDisconnect_triggered();

    void on_actionRefresh_triggered();

    void on_pb_add_operator_clicked();

    void on_pb_operator_clicked();

    void on_table_admin_clicked(const QModelIndex &index);

    void on_table_card_clicked(const QModelIndex &index);

    void on_pb_delete_operator_clicked();

    void on_pb_add_card_clicked();

    void on_pb_add_saldo_clicked();

    void on_pb_smartcard_clicked();

    void on_le_rfid_textChanged(const QString &arg1);

    void on_pb_cancel_smartcard_clicked();

    void on_pb_cancel_operator_clicked();

private:
    Ui::MainWindow *ui;

    struct setting s;
    SettingsDialog *settingsDialog;
    bool set;

    QSerialPort *serial_port; bool serial_connect; QString
str_read;
    QSqlDatabase db;
    init_mysql initMysql;
    bool database_connect;
    int id_user;

    struct opt adm;

```

```

    int curr_adm;
//    int jml_admin;
    struct card crd;
    int curr_crd;
//    int jml_card;
    struct log_history logHistory;
//    int jml_log;

    void setDefault(bool b);
    void setOperator();
    void getOperator();
    void setCard();
    void getCard();
    void setDataLogHistory();
    void getDataLogHistory();
    void getSetting();

    void addOperator(QString username, QString password);
    void deleteOperator(int id_operator);
    void addCard(int rfid, int saldo);
    void addSaldo(int rfid, int saldo, QString action, int mode);
    QString getSaldo(int rfid);
    bool processCard(int rfid, int price);
};

#endif // MAINWINDOW_H

```

Settingdialog.h :

```

#ifndef SETTINGSDIALOG_H
#define SETTINGSDIALOG_H

#include <QDialog>
#include "global.h"

namespace Ui {
    class SettingsDialog;
}

/**
 * @brief Deklarasi Class Object SettingsDialog
 */
class SettingsDialog : public QDialog
{
    Q_OBJECT

public:
    /**
     * @brief Deklarasi Constructor SettingsDialog
     * @param parent
     */
    explicit SettingsDialog(QWidget *parent = 0);
    ~SettingsDialog();

private:

```

```

/**
 * @brief Deklarasi Object ui (User Interface)
 */
Ui::SettingsDialog *ui;

/**
 * @brief Deklarasi Struct s
 */
struct setting s;

/**
 * @brief Deklarasi Procedure fillPortsInfo
 */
void fillPortsInfo();
/**
 * @brief Deklarasi Procedure fillPortsParameters
 */
void fillPortsParameters();
/**
 * @brief Deklarasi Procedure setSetting
 */
void setSetting();
/**
 * @brief Deklarasi Procedure getSetting
 */
void getSetting();
/**
 * @brief Deklarasi Procedure readSetting
 */
void readSetting();
/**
 * @brief Deklarasi Procedure writeSetting
 */
void writeSetting();

private slots:
/**
 * @brief Deklarasi Procedure showPortInfo
 * @param idx
 */
void showPortInfo(int idx);
/**
 * @brief Deklarasi Procedure on_pbApply_clicked
 */
void on_pbApply_clicked();
/**
 * @brief Deklarasi Procedure on_pbCancel_clicked
 */
void on_pbCancel_clicked();
};

#endif // SETTINGSDIALOG_H

```

Init_mysql.cpp :

```

#include "init_mysql.h"

init_mysql::init_mysql()
{
}

QSqlDatabase init_mysql::connect_db() {

    QFile db_path("settings.ini");
    if (db_path.exists()) {
        QSettings db_sett("settings.ini", QSettings::IniFormat);
        host = db_sett.value("DATABASE/HOST").toString();
        db_name = db_sett.value("DATABASE/DB_NAME").toString();
        user_name = db_sett.value("DATABASE/USERNAME").toString();
        password = db_sett.value("DATABASE/PASSWORD").toString();

        QSqlDatabase db = QSqlDatabase::addDatabase("QMYSQL");
        db.setHostName(host);
        db.setDatabaseName(db_name);
        db.setUserName(user_name);
        db.setPassword(password);

        return (QSqlDatabase) db;
    } else {
        host = "localhost";
        db_name = "emoney";
        user_name = "root";
        password = "root";

        QSettings db_sett("settings.ini", QSettings::IniFormat);
        db_sett.beginGroup("DATABASE");
        db_sett.setValue("HOST", host.toUtf8());
        db_sett.setValue("DB_NAME", db_name.toUtf8());
        db_sett.setValue("USERNAME", user_name.toUtf8());
        db_sett.setValue("PASSWORD", password.toUtf8());
        db_sett.endGroup();

        QSqlDatabase db = QSqlDatabase::addDatabase("QMYSQL");
        db.setHostName(host);
        db.setDatabaseName(db_name);
        db.setUserName(user_name);
        db.setPassword(password);

        return (QSqlDatabase) db;
    }
}

void init_mysql::close(QSqlDatabase db)
{
    QString connection;
    connection = db.connectionName();
    db.close();
    db = QSqlDatabase();
    db.removeDatabase(connection);
}

```

logindialog.cpp :

```

#include "logindialog.h"
#include "ui_logindialog.h"

LoginDialog::LoginDialog(QWidget *parent) :
    QDialog(parent),
    ui(new Ui::LoginDialog)
{
    ui->setupUi(this);

    db = mysql.connect_db();
    db.open();
    if (!db.isOpen()) {
        QMessageBox::critical(this, "Database", "Database is not
open ...!!");
        this->close();
    } else {
        this->getAdmin();
    }
}

LoginDialog::~LoginDialog()
{
    delete ui;
}

void LoginDialog::on_pushButton_clicked()
{
    db.open();
    getAdmin();
    bool ok = false;
    int user_id;
    for (int i = 0; i < user.length(); i+=3) {
        if (ui->lineEdit_username->text() == user.at(i+1) && ui-
>lineEdit_password->text() == user.at(i+2)) {
            user_id = user.at(i).toInt();
            ok = true;
            break;
        } else {
            ok = false;
        }
    }

    if (ok) {
        mWindow = new MainWindow(this, user_id);
        mWindow->show();
        this->hide();
    } else {
        QMessageBox::critical(this, "Login", "Wrong Username or
Password !!!");
    }
}

void LoginDialog::getAdmin() {
    user.clear();
    QSqlQuery q(db);

```

```

    q.prepare("SELECT id, username, password FROM admin WHERE
status = 1");
    if (!q.exec()) {
        return;
    }
    else {
        while (q.next()) {
            user.append(q.value(0).toString().toLatin1());
            user.append(q.value(1).toString().toLatin1());
            user.append(q.value(2).toString().toLatin1());
        }
    }
    q.finish();
    db.close();
    return;
}

```

main.cpp :

```

#include "logindialog.h"
#include <QApplication>

int main(int argc, char *argv[])
{
    QApplication a(argc, argv);
    LoginDialog utama;
    utama.show();

    return a.exec();
}

```

mainwindow.cpp

```

#include "mainwindow.h"
#include "ui_mainwindow.h"

MainWindow::MainWindow(QWidget *parent, int user_id) :
    QMainWindow(parent),
    ui(new Ui::MainWindow)
{
    ui->setupUi(this);
    MainWindow::setWindowTitle("Emoney Management Software");

    id_user = user_id;
    ui->tabWidget->setCurrentIndex(0);
    this->setDefault(true);

    curr_adm = 0;
    curr_crd = 0;

    ui->stackedWidget->setCurrentIndex(0);
}

```

```

    if (id_user == 1) {
        ui->pb_add_operator->show();
        ui->pb_delete_operator->show();
    } else {
        ui->pb_add_operator->hide();
        ui->pb_delete_operator->hide();
    }
}

MainWindow::~MainWindow()
{
    delete ui;
}

void MainWindow::setDefault(bool b) {
    ui->actionConnect->setEnabled(b);
    ui->actionDisconnect->setEnabled(!b);
    ui->actionRefresh->setEnabled(!b);
    ui->tabWidget->setEnabled(!b);
}

void MainWindow::on_actionRefresh_triggered()
{
    this->getOperator();
    this->setOperator();

    this->getCard();
    this->setCard();

    this->getDataLogHistory();
    this->setDataLogHistory();

    ui->le_username->clear();
    ui->le_password->clear();

    ui->le_rfid->clear();
    ui->le_saldo->clear();
}

void MainWindow::on_actionConnect_triggered()
{
    if (!set || this->s.serialCommunication.name == "N/A" || this-
>s.serialCommunication.name == "" || this->s.Database.host == "")
    {
        settingsDialog = new SettingsDialog;
        settingsDialog->exec();
        getSetting();
    }
    if (this->s.serialCommunication.name != "N/A" || this-
>s.Database.host != "") {
        serial_port = new QSerialPort;
        connect(serial_port, SIGNAL(readyRead()), this,
SLOT(readData_Serial()));
        serial_port->setPortName(this-
>s.serialCommunication.name);
        serial_port->setBaudRate(this-
>s.serialCommunication.baudRate);
        serial_port->setDataBits(this-
>s.serialCommunication.dataBits);
    }
}

```



```

        serial_port->setParity(this-
>s.serialCommunication.parity);
        serial_port->setStopBits(this-
>s.serialCommunication.stopBits);
        serial_port->setFlowControl(this-
>s.serialCommunication.flowControl);
        if (serial_port->open(QIODevice::ReadWrite)) {
            serial_connect = true;
        } else {
            serial_connect = false;
            serial_port = NULL;
        }
        db = initMysql.connect_db();
        db.open();
        if (db.isOpen()) {
            database_connect = true;
            this->on_actionRefresh_triggered();
        } else {
            database_connect = false;
        }
    } else {
        set = false;
        serial_connect = false;
    }
}

if (serial_connect && database_connect) {
    QMessageBox::information(this, "Serial Communication &
Database", " Berhasil Terkoneksi");
    this->setDefault(false);
} else {
    if (!serial_connect && database_connect) {
        QMessageBox::critical(this, "Serial Communication &
Database", "Koneksi Serial Gagal !!");
        this->setDefault(false);
    } else if (serial_connect && !database_connect) {
        QMessageBox::critical(this, "Serial Communication &
Database", "Koneksi Database Gagal !!");
        this->setDefault(true);
    } else {
        QMessageBox::critical(this, "Serial Communication &
Database", "Koneksi Serial dan Database Gagal !!");
        this->setDefault(true);
    }
}
}

void MainWindow::on_actionDisconnect_triggered()
{
    if (serial_connect && database_connect) {
        if (serial_port->isOpen()) {
            serial_port->close();
            serial_connect = false;
        }
        if (db.isOpen()) {
            db.close();
            database_connect = false;
        }
        if (serial_connect && database_connect) {

```

```

        QMessageBox::critical(this, "Serial Communication &
Database", "Serial dan Database gagal Disconnect");
    } else {
        if (!serial_connect && database_connect) {
            QMessageBox::critical(this, "Serial Communication
& Database", "Serial Berhasil Disconnect/nDatabase gagal
Disconnect");
        } else if (serial_connect && !database_connect) {
            QMessageBox::critical(this, "Serial Communication
& Database", "Serial gagal Disconnect/nDatabase Berhasil
Disconnect");
        } else {
            QMessageBox::information(this, "Serial
Communication & Database", "Serial dan Database Berhasil
Disconnect");
            this->setDefault(true);
        }
    }
} else {
    QMessageBox::warning(this, "Serial Communication", "Tidak
ada Koneksi");
    this->setDefault(true);
}
}

void MainWindow::on_actionSetting_triggered()
{
    settingsDialog = new SettingsDialog;
    settingsDialog->exec();
}

void MainWindow::readData_Serial()
{
    str_read.append(serial_port->readAll());
    if (str_read.indexOf("GET") > 0 && str_read.indexOf("#") > 0
&& (str_read.indexOf("*GET;") < str_read.indexOf("#"))) {
        int a = str_read.indexOf("GET;");
        int b = str_read.indexOf("#");
        str_read = str_read.mid(a, b-a);

        str_read.remove("GET;").remove("#").remove("\r\n").remove("\r").re
move("\n");
        serial_port->write(QByteArray("GET;Your Saldo ;;Rp"+this-
>getSaldo(str_read.toInt()).toUtf8()+"#"));
        str_read.clear();
    } else if (str_read.indexOf("SET;") > 0 &&
str_read.indexOf("#") > 0 && (str_read.indexOf("*SET;") <
str_read.indexOf("#"))) {
        int a = str_read.indexOf("SET;");
        int b = str_read.indexOf("#");
        str_read = str_read.mid(a, b-a);

        str_read.remove("SET;").remove("#").remove("\r\n").remove("\r").re
move("\n");
        QStringList list_temp = str_read.split(";");
        if (this->getSaldo(list_temp.at(0).toInt()).toInt() >=
list_temp.at(1).toInt()) {
            if (this->processCard(list_temp.at(0).toInt(),
list_temp.at(1).toInt())) {

```

```

        serial_port->write(QByteArray("SET;SUCCESS;1#"));
    } else {
        serial_port->write(QByteArray("SET;FAIL;0#"));
    }
} else {
    serial_port->write(QByteArray("SET;FAIL;0#"));
}
str_read.clear();
}
}

void MainWindow::getSetting()
{
    if (QFile("settings.ini").exists()) {
        QSettings sett("settings.ini", QSettings::IniFormat);

        this->s.serialCommunication.name =
sett.value("SERIAL/NAME").toString();
        this->s.serialCommunication.c_name_id =
sett.value("SERIAL/C_NAME_ID").toInt();

        this->s.serialCommunication.baudRate =
static_cast<QSerialPort::BaudRate>(
            sett.value("SERIAL/BAUDRATE").toInt());
        this->s.serialCommunication.c_baudRate_id =
sett.value("SERIAL/C_BAUDRATE_ID").toInt();
        this->s.serialCommunication.stringBaudRate =
sett.value("SERIAL/STRING_BAUDRATE").toString();

        this->s.serialCommunication.dataBits =
static_cast<QSerialPort::DataBits>(
            sett.value("SERIAL/DATA_BITS").toInt());
        this->s.serialCommunication.c_dataBits_id =
sett.value("SERIAL/C_DATA_BITS_ID").toInt();
        this->s.serialCommunication.stringDataBits =
sett.value("SERIAL/STRING_DATA_BITS").toString();

        this->s.serialCommunication.parity =
static_cast<QSerialPort::Parity>(
            sett.value("SERIAL/PARITY").toInt());
        this->s.serialCommunication.c_parity_id =
sett.value("SERIAL/C_PARITY_ID").toInt();
        this->s.serialCommunication.stringParity =
sett.value("SERIAL/STRING_PARITY").toString();

        this->s.serialCommunication.stopBits =
static_cast<QSerialPort::StopBits>(
            sett.value("SERIAL/STOP_BITS").toInt());
        this->s.serialCommunication.c_stopBits_id =
sett.value("SERIAL/C_STOP_BITS_ID").toInt();
        this->s.serialCommunication.stringStopBits =
sett.value("SERIAL/STRING_STOP_BITS").toString();

        this->s.serialCommunication.flowControl =
static_cast<QSerialPort::FlowControl>(
            sett.value("SERIAL/FLOW_CONTROL").toInt());
        this->s.serialCommunication.c_flowControl_id =
sett.value("SERIAL/C_FLOW_CONTROL_ID").toInt();
    }
}

```

```

        this->s.serialCommunication.stringFlowControl =
settt.value("SERIAL/STRING_FLOW_CONTROL").toString();

        this->s.Database.host =
settt.value("DATABASE/HOST").toString();
        this->s.Database.db_name =
settt.value("DATABASE/DB_NAME").toString();
        this->s.Database.username =
settt.value("DATABASE/USERNAME").toString();
        this->s.Database.password =
settt.value("DATABASE/PASSWORD").toString();

        if (!this->s.serialCommunication.name.isEmpty() && !this-
>s.Database.host.isEmpty()) {
            set = true;
        } else {
            set = false;
        }
    } else {
        set = false;
    }
}

void MainWindow::setOperator()
{
    QStandardItemModel *Model_Operator = new
QStandardItemModel(adm.jml_operator, 2, this);
    // Model_Operator->setHorizontalHeaderItem(0, new
QStandardItem(QString("No")));
    Model_Operator->setHorizontalHeaderItem(0, new
QStandardItem(QString("ID")));
    Model_Operator->setHorizontalHeaderItem(1, new
QStandardItem(QString("Username")));

    ui->table_admin->setModel(Model_Operator);

    for (int i = 0; i < adm.jml_operator; i++) {
    // Model_Operator->setItem(i, 0, new
QStandardItem(QString::number(i+1)));
        Model_Operator->setItem(i, 0, new
QStandardItem(adm.id.at(i)));
        Model_Operator->setItem(i, 1, new
QStandardItem(adm.username.at(i)));
    }
}

void MainWindow::getOperator()
{
    QSqlQuery q(db);
    if (q.exec("call get_operator()")) {
        adm.id.clear(); adm.username.clear(); adm.jml_operator =
0;
        while (q.next()) {
            adm.id.append(q.value(0).toString());
            adm.username.append(q.value(1).toString());
            adm.jml_operator++;
        }
    } else {

```

```

        QMessageBox::warning(this, "Database", "Database Error
...");
    }
}

void MainWindow::setCard()
{
    QStandardItemModel *Model_Card = new
QStandardItemModel(crd.jml_card, 2, this);
//    Model_Card->setHorizontalHeaderItem(0, new
QStandardItem(QString("No")));
    Model_Card->setHorizontalHeaderItem(0, new
QStandardItem(QString("RFID")));
    Model_Card->setHorizontalHeaderItem(1, new
QStandardItem(QString("Saldo")));

    ui->table_card->setModel(Model_Card);

    for (int i = 0; i < crd.jml_card; i++) {
//        Model_Card->setItem(i, 0, new
QStandardItem(QString::number(i+1)));
        Model_Card->setItem(i, 0, new
QStandardItem(crd.rfid.at(i)));
        Model_Card->setItem(i, 1, new
QStandardItem(crd.saldo.at(i)));
    }
}

void MainWindow::getCard()
{
    QSqlQuery q(db);
    if (q.exec("call get_card()")) {
        crd.rfid.clear(); crd.saldo.clear(); crd.jml_card = 0;
        while (q.next()) {
            crd.rfid.append(q.value(0).toString());
            crd.saldo.append(q.value(1).toString());
            crd.jml_card++;
        }
    } else {
        QMessageBox::warning(this, "Database", "Database Error
...");
    }
}

void MainWindow::setDataLogHistory()
{
    QStandardItemModel *Model_Log_History = new
QStandardItemModel(logHistory.jml_log, 4, this);
//    Model_Log_History->setHorizontalHeaderItem(0, new
QStandardItem(QString("No")));
    Model_Log_History->setHorizontalHeaderItem(0, new
QStandardItem(QString("Action")));
    Model_Log_History->setHorizontalHeaderItem(1, new
QStandardItem(QString("Operator ID")));
    Model_Log_History->setHorizontalHeaderItem(2, new
QStandardItem(QString("Card ID")));
    Model_Log_History->setHorizontalHeaderItem(3, new
QStandardItem(QString("Date")));
}

```

```

    ui->table_log->setModel(Model_Log_History);
    ui->table_log->setColumnWidth(0, 400);
    ui->table_log->setColumnWidth(1, 70);
    ui->table_log->setColumnWidth(2, 70);
    ui->table_log->setColumnWidth(3, 150);

    for (int i = 0; i < logHistory.jml_log; i++) {
//      Model_Log_History->setItem(i, 0, new
QStandardItem(QString::number(i+1)));
        Model_Log_History->setItem(i, 0, new
QStandardItem(logHistory.action.at(i)));
        Model_Log_History->setItem(i, 1, new
QStandardItem(logHistory.user_id.at(i)));
        Model_Log_History->setItem(i, 2, new
QStandardItem(logHistory.card_id.at(i)));
        Model_Log_History->setItem(i, 3, new
QStandardItem(logHistory.date.at(i)));
    }
}

void MainWindow::getDataLogHistory()
{
    QSqlQuery q(db);
    if (q.exec("call get_log()")) {
        logHistory.action.clear(); logHistory.user_id.clear();
logHistory.card_id.clear(); logHistory.date.clear();
logHistory.jml_log = 0;
        while (q.next()) {
            logHistory.action.append(q.value(0).toString());
logHistory.user_id.append(q.value(1).toString());
logHistory.card_id.append(q.value(2).toString());

logHistory.date.append(q.value(3).toString().replace("T", " |
"));
            logHistory.jml_log++;
        }
    } else {
        QMessageBox::warning(this, "Database", "Database Error
...");
    }
}

void MainWindow::on_pb_add_operator_clicked()
{
    ui->stackedWidget->setCurrentIndex(1);
    ui->pb_operator->setText("Add Operator");
}

void MainWindow::on_pb_operator_clicked()
{
    if (ui->pb_operator->text() == "Add Operator") {
        if (!ui->le_username->text().isEmpty() && !ui-
>le_password->text().isEmpty()) {
            addOperator(ui->le_username->text(), ui->le_password-
>text());
            this->on_actionRefresh_triggered();
        } else {
            QMessageBox::warning(this, "Operator", "Username or
Password must be null ..");
        }
    }
}

```

```

    }
}
ui->stackedWidget->setCurrentIndex(0);
}

void MainWindow::addOperator(QString username, QString password) {
    QStringList result;
    QSqlQuery q(db);
    if (q.exec("call
insert_operator("+QString::number(id_user)+"','"+username+"','"+pas
sword+"',@result)")) {
        while (q.next()) {
            result.append(q.value(0).toString());
            result.append(q.value(1).toString());
        }
        if (result.at(0).toInt()) {
            QMessageBox::information(this, "Operator",
result.at(1));
        } else {
            QMessageBox::critical(this, "Operator", result.at(1));
        }
    } else {
        QMessageBox::warning(this, "Database", "Database Error
...");
    }
}

void MainWindow::on_table_admin_clicked(const QModelIndex &index)
{
    curr_adm = index.row();
    curr_adm++;
}

void MainWindow::on_table_card_clicked(const QModelIndex &index)
{
    curr_crd = index.row();
    curr_crd++;
}

void MainWindow::on_pb_delete_operator_clicked()
{
    if (curr_adm > 1) {
        if (id_user != adm.id.at(curr_adm-1).toInt()) {
            deleteOperator(adm.id.at(curr_adm-1).toInt());
            this->on_actionRefresh_triggered();
        }
    } else {
        QMessageBox::warning(this, "Operator", "Please choose one
operator for deleting ..");
    }
}

void MainWindow::deleteOperator(int id_operator) {
    QStringList result;
    QSqlQuery q(db);
    if (q.exec("call
delete_operator("+QString::number(id_user)+"','"+QString::number(id_
operator)+"',@result)")) {
        while (q.next()) {

```

```

        result.append(q.value(0).toString());
        result.append(q.value(1).toString());
    }
    if (result.at(0).toInt()) {
        QMessageBox::information(this, "Operator",
result.at(1));
    } else {
        QMessageBox::critical(this, "Operator", result.at(1));
    }
} else {
    QMessageBox::warning(this, "Database", "Database Error
...");
}
}

void MainWindow::on_pb_add_card_clicked()
{
    ui->stackedWidget->setCurrentIndex(2);
    ui->pb_smartcard->setText("Add Card");
}

void MainWindow::on_pb_add_saldo_clicked()
{
    ui->stackedWidget->setCurrentIndex(2);
    ui->pb_smartcard->setText("Add Saldo");
//    ui->le_rfid->setText(crd.rfid.at(curr_crd-1));
//    ui->le_rfid->setEnabled(false);
}

void MainWindow::on_pb_smartcard_clicked()
{
    if (ui->pb_smartcard->text() == "Add Card") {
        if (!ui->le_rfid->text().isEmpty()) {
            int saldo = ui->le_saldo->text().toInt();
            if (ui->rb_10rb->isChecked()) saldo = saldo + 10000;
            if (ui->rb_20rb->isChecked()) saldo = saldo + 20000;
            if (ui->rb_50rb->isChecked()) saldo = saldo + 50000;
            if (ui->rb_100rb->isChecked()) saldo = saldo + 100000;
            this->addCard(ui->le_rfid->text().toInt(), saldo);
            this->on_actionRefresh_triggered();
        }
    } else if (ui->pb_smartcard->text() == "Add Saldo") {
        bool ok = false;
        for (int i = 0; i < crd.jml_card; i++) {
            if (ui->le_rfid->text().toInt() ==
crd.rfid.at(i).toInt()) {
                ok = true;
                break;
            }
        }
        if (ok) {
            int saldo = ui->le_saldo->text().toInt();
            if (ui->rb_10rb->isChecked()) saldo = saldo + 10000;
            if (ui->rb_20rb->isChecked()) saldo = saldo + 20000;
            if (ui->rb_50rb->isChecked()) saldo = saldo + 50000;
            if (ui->rb_100rb->isChecked()) saldo = saldo + 100000;
            this->addSaldo(ui->le_rfid->text().toInt(),

```



```

        saldo,
        "Berhasil menambahkan saldo sebesar "+
        QString::number(saldo-ui->le_saldo-
>text().toInt()+
        " ke RFID = ",
        0);
        this->on_actionRefresh_triggered();
    } else {
        QMessageBox::critical(this, "Card", "Kartu belum
terdaftar ..");
    }
    ui->stackedWidget->setCurrentIndex(0);
}

void MainWindow::addCard(int rfid, int saldo)
{
    QStringList result;
    QSqlQuery q(db);
    if (q.exec("call
insert_card(QString::number(id_user)+", "+QString::number(rfid)+"
, "+QString::number(saldo)+", @result)")) {
        while (q.next()) {
            result.append(q.value(0).toString());
            result.append(q.value(1).toString());
        }
        if (result.at(0).toInt()) {
            QMessageBox::information(this, "Card", result.at(1));
        } else {
            QMessageBox::critical(this, "Card", result.at(1));
        }
    } else {
        QMessageBox::warning(this, "Database", "Database Error
...");
    }
}

void MainWindow::addSaldo(int rfid, int saldo, QString action, int
mode)
{
    QStringList result;
    QSqlQuery q(db);
    if (q.exec("call
update_saldo(QString::number(id_user)+", "+QString::number(rfid)+"
, "+QString::number(saldo)+", '"+action+"', "+QString::number(mode)+"
, @result)")) {
        while (q.next()) {
            result.append(q.value(0).toString());
            result.append(q.value(1).toString());
        }
        if (result.at(0).toInt()) {
            QMessageBox::information(this, "Card", result.at(1));
        } else {
            QMessageBox::critical(this, "Card", result.at(1));
        }
    } else {
        QMessageBox::warning(this, "Database", "Database Error
...");
    }
}

```

```

}

QString MainWindow::getSaldo(int rfid)
{
    QString result;
    QSqlQuery q(db);
    if (q.exec("call get_saldo("+QString::number(rfid)+")")) {
        while (q.next()) {
            result = q.value(0).toString();
        }
    } else {
        QMessageBox::warning(this, "Database", "Database Error
...");
    }
    return result;
}

bool MainWindow::processCard(int rfid, int price) {
    bool result;
    QSqlQuery q(db);
    if (q.exec("call
process_card("+QString::number(rfid)+","+QString::number(price)+")
")) {
        result = true;
    } else {
        result = false;
    }
    return result;
}

void MainWindow::on_le_rfid_textChanged(const QString &arg1)
{
    //    if (!arg1.isEmpty() && arg1.indexOf("\n") > 0) {
        if (ui->pb_smartcard->text() == "Add Saldo") {
            bool ok = false;
            for (int i = 0; i < crd.jml_card; i++) {
                if (arg1.toInt() == crd.rfid.at(i).toInt()) {
                    ui->le_saldo->setText(crd.saldo.at(i));
                    ok = true;
                    break;
                }
            }
            if (!ok) {
                ui->le_saldo->clear();
            }
        }
    //    }
}

void MainWindow::on_pb_cancel_smartcard_clicked()
{
    ui->stackedWidget->setCurrentIndex(0);
}

void MainWindow::on_pb_cancel_operator_clicked()
{
    ui->stackedWidget->setCurrentIndex(0);
}

```

Settingsdialog.cpp :

```

#include "settingsdialog.h"
#include "ui_settingsdialog.h"

SettingsDialog::SettingsDialog(QWidget *parent) :
    QDialog(parent),
    ui(new Ui::SettingsDialog)
{
    ui->setupUi(this);

    connect(ui->serialPortInfoListBox,
        SIGNAL(currentIndexChanged(int)), this, SLOT(showPortInfo(int)));

    ui->tabWidget->setCurrentIndex(0);

    fillPortsInfo();
    fillPortsParameters();

    readSetting();
}

SettingsDialog::~SettingsDialog()
{
    delete ui;
}

void SettingsDialog::fillPortsInfo()
{
    ui->serialPortInfoListBox->clear();
    QString description;
    QString manufacturer;
    QString serialNumber;
    foreach (const QSerialPortInfo &info,
        QSerialPortInfo::availablePorts()) {
        QStringList list;
        description = info.description();
        manufacturer = info.manufacturer();
        serialNumber = info.productIdentifier();
        list << info.portName()
            << (!description.isEmpty() ? description : "N/A")
            << (!manufacturer.isEmpty() ? manufacturer : "N/A")
            << (!serialNumber.isEmpty() ? serialNumber : "N/A")
            << info.systemLocation()
            << (info.vendorIdentifier() ?
                QString::number(info.vendorIdentifier(), 16) : "N/A")
            << (info.productIdentifier() ?
                QString::number(info.productIdentifier(), 16) : "N/A");

        ui->serialPortInfoListBox->addItem(list.first(), list);
    }

    ui->serialPortInfoListBox->addItem(tr("N/A"));
}

```

```

}

void SettingsDialog::fillPortsParameters()
{
    ui->baudRateBox->addItem(QStringLiteral("9600"),
    QSerialPort::Baud9600);
    ui->baudRateBox->addItem(QStringLiteral("19200"),
    QSerialPort::Baud19200);
    ui->baudRateBox->addItem(QStringLiteral("38400"),
    QSerialPort::Baud38400);
    ui->baudRateBox->addItem(QStringLiteral("57600"),
    QSerialPort::Baud57600);
    ui->baudRateBox->addItem(QStringLiteral("115200"),
    QSerialPort::Baud115200);
    ui->baudRateBox->addItem(tr("Custom"));

    ui->dataBitsBox->addItem(QStringLiteral("5"),
    QSerialPort::Data5);
    ui->dataBitsBox->addItem(QStringLiteral("6"),
    QSerialPort::Data6);
    ui->dataBitsBox->addItem(QStringLiteral("7"),
    QSerialPort::Data7);
    ui->dataBitsBox->addItem(QStringLiteral("8"),
    QSerialPort::Data8);
    ui->dataBitsBox->setCurrentIndex(3);

    ui->parityBox->addItem(tr("None"), QSerialPort::NoParity);
    ui->parityBox->addItem(tr("Even"), QSerialPort::EvenParity);
    ui->parityBox->addItem(tr("Odd"), QSerialPort::OddParity);
    ui->parityBox->addItem(tr("Mark"), QSerialPort::MarkParity);
    ui->parityBox->addItem(tr("Space"), QSerialPort::SpaceParity);

    ui->stopBitsBox->addItem(QStringLiteral("1"),
    QSerialPort::OneStop);
#ifdef Q_OS_WIN
    ui->stopBitsBox->addItem(tr("1.5"),
    QSerialPort::OneAndHalfStop);
#endif
    ui->stopBitsBox->addItem(QStringLiteral("2"),
    QSerialPort::TwoStop);

    ui->flowControlBox->addItem(tr("None"),
    QSerialPort::NoFlowControl);
    ui->flowControlBox->addItem(tr("RTS/CTS"),
    QSerialPort::HardwareControl);
    ui->flowControlBox->addItem(tr("XON/XOFF"),
    QSerialPort::SoftwareControl);
}

void SettingsDialog::showPortInfo(int idx)
{
    if (idx == -1)
        return;

    QStringList list = ui->serialPortInfoListBox-
    >itemData(idx).toStringList();
    ui->descriptionLabel->setText(tr("Description:
    %1").arg(list.count() > 1 ? list.at(1) : "N/A"));
}

```

```

        ui->manufacturerLabel->setText(tr("Manufacturer:
%1").arg(list.count() > 2 ? list.at(2) : "N/A"));
        ui->serialNumberLabel->setText(tr("Serial number:
%1").arg(list.count() > 3 ? list.at(3) : "N/A"));
        ui->locationLabel->setText(tr("Location: %1").arg(list.count()
> 4 ? list.at(4) : "N/A"));
        ui->vidLabel->setText(tr("Vendor Identifier:
%1").arg(list.count() > 5 ? list.at(5) : "N/A"));
        ui->pidLabel->setText(tr("Product Identifier:
%1").arg(list.count() > 6 ? list.at(6) : "N/A"));
    }

void SettingsDialog::on_pbApply_clicked()
{
    writeSetting();
    //    this->close();
    this->hide();
}

void SettingsDialog::on_pbCancel_clicked()
{
    //    this->close();
    this->hide();
}

void SettingsDialog::setSetting()
{
    for (int i = ui->serialPortInfoListBox->count(); i >= 0; i--)
    {
        ui->serialPortInfoListBox->setCurrentIndex(i);
        if (ui->serialPortInfoListBox->currentText() == this-
>s.serialCommunication.name) {
            break;
        }
    }
    ui->baudRateBox->setCurrentIndex(this-
>s.serialCommunication.c_baudRate_id);
    ui->dataBitsBox->setCurrentIndex(this-
>s.serialCommunication.c_dataBits_id);
    ui->parityBox->setCurrentIndex(this-
>s.serialCommunication.c_parity_id);
    ui->stopBitsBox->setCurrentIndex(this-
>s.serialCommunication.c_stopBits_id);
    ui->flowControlBox->setCurrentIndex(this-
>s.serialCommunication.c_flowControl_id);

    ui->le_db_host->setText(this->s.Database.host);
    ui->le_db_name->setText(this->s.Database.db_name);
    ui->le_db_username->setText(this->s.Database.username);
    ui->le_db_password->setText(this->s.Database.password);
}

void SettingsDialog::getSetting()
{
    this->s.serialCommunication.name = ui->serialPortInfoListBox-
>currentText();
    this->s.serialCommunication.c_name_id = ui-
>serialPortInfoListBox->currentIndex();
}

```

```

        this->s.serialCommunication.baudRate =
static_cast<QSerialPort::BaudRate>(
            ui->baudRateBox->currentData().toInt());
        this->s.serialCommunication.c_baudRate_id = ui->baudRateBox-
>currentIndex();
        this->s.serialCommunication.stringBaudRate = ui->baudRateBox-
>currentText();

        this->s.serialCommunication.dataBits =
static_cast<QSerialPort::DataBits>(
            ui->dataBitsBox->currentData().toInt());
        this->s.serialCommunication.c_dataBits_id = ui->dataBitsBox-
>currentIndex();
        this->s.serialCommunication.stringDataBits = ui->dataBitsBox-
>currentText();

        this->s.serialCommunication.parity =
static_cast<QSerialPort::Parity>(
            ui->parityBox->currentData().toInt());
        this->s.serialCommunication.c_parity_id = ui->parityBox-
>currentIndex();
        this->s.serialCommunication.stringParity = ui->parityBox-
>currentText();

        this->s.serialCommunication.stopBits =
static_cast<QSerialPort::StopBits>(
            ui->stopBitsBox->currentData().toInt());
        this->s.serialCommunication.c_stopBits_id = ui->stopBitsBox-
>currentIndex();
        this->s.serialCommunication.stringStopBits = ui->stopBitsBox-
>currentText();

        this->s.serialCommunication.flowControl =
static_cast<QSerialPort::FlowControl>(
            ui->flowControlBox->currentData().toInt());
        this->s.serialCommunication.c_flowControl_id = ui-
>flowControlBox->currentIndex();
        this->s.serialCommunication.stringFlowControl = ui-
>flowControlBox->currentText();

        this->s.Database.host = ui->le_db_host->text();
        this->s.Database.db_name = ui->le_db_name->text();
        this->s.Database.username = ui->le_db_username->text();
        this->s.Database.password = ui->le_db_password->text();
    }

void SettingsDialog::readSetting()
{
    if (QFile("settings.ini").exists()) {
        QSettings sett("settings.ini", QSettings::IniFormat);

        this->s.serialCommunication.name =
sett.value("SERIAL/NAME").toString();
        this->s.serialCommunication.c_baudRate_id =
sett.value("SERIAL/C_BAUDRATE_ID").toInt();
        this->s.serialCommunication.c_dataBits_id =
sett.value("SERIAL/C_DATA_BITS_ID").toInt();
        this->s.serialCommunication.c_parity_id =
sett.value("SERIAL/C_PARITY_ID").toInt();
    }
}

```

```

        this->s.serialCommunication.c_stopBits_id =
settt.value("SERIAL/C_STOP_BITS_ID").toInt();
        this->s.serialCommunication.c_flowControl_id =
settt.value("SERIAL/C_FLOW_CONTROL_ID").toInt();

        this->s.Database.host =
settt.value("DATABASE/HOST").toString();
        this->s.Database.db_name =
settt.value("DATABASE/DB_NAME").toString();
        this->s.Database.username =
settt.value("DATABASE/USERNAME").toString();
        this->s.Database.password =
settt.value("DATABASE/PASSWORD").toString();

        setSetting();
    }
}

void SettingsDialog::writeSetting()
{
    getSetting();
    QSettings sett("settings.ini", QSettings::IniFormat);

    /** SERIAL COMMUNICATION **/

    sett.beginGroup("SERIAL");
    sett.setValue("NAME", this->s.serialCommunication.name);
    sett.setValue("C_NAME_ID", this->
>s.serialCommunication.c_name_id);

    sett.setValue("BAUDRATE", this->
>s.serialCommunication.baudRate);
    sett.setValue("C_BAUDRATE_ID", this->
>s.serialCommunication.c_baudRate_id);
    sett.setValue("STRING_BAUDRATE", this->
>s.serialCommunication.stringBaudRate);

    sett.setValue("DATA_BITS", this->
>s.serialCommunication.dataBits);
    sett.setValue("C_DATA_BITS_ID", this->
>s.serialCommunication.c_dataBits_id);
    sett.setValue("STRING_DATA_BITS", this->
>s.serialCommunication.stringDataBits);

    sett.setValue("PARITY", this->s.serialCommunication.parity);
    sett.setValue("C_PARITY_ID", this->
>s.serialCommunication.c_parity_id);
    sett.setValue("STRING_PARITY", this->
>s.serialCommunication.stringParity);

    sett.setValue("STOP_BITS", this->
>s.serialCommunication.stopBits);
    sett.setValue("C_STOP_BITS_ID", this->
>s.serialCommunication.c_stopBits_id);
    sett.setValue("STRING_STOP_BITS", this->
>s.serialCommunication.stringStopBits);

    sett.setValue("FLOW_CONTROL", this->
>s.serialCommunication.flowControl);

```

```
    sett.setValue("C_FLOW_CONTROL_ID", this->s.serialCommunication.c_flowControl_id);
    sett.setValue("STRING_FLOW_CONTROL", this->s.serialCommunication.stringFlowControl);
    sett.endGroup();

    /*** DATABASE ***/

    sett.beginGroup("DATABASE");
    sett.setValue("HOST", this->s.Database.host);
    sett.setValue("DB_NAME", this->s.Database.db_name);
    sett.setValue("USERNAME", this->s.Database.username);
    sett.setValue("PASSWORD", this->s.Database.password);
    sett.endGroup();
}
```

DAFTAR RIWAYAT HIDUP



Peneliti bernama lengkap Riandy Heryanda, lahir di Jakarta 29 Juli 1992, merupakan anak keempat dari pasangan Tjatja Surianda dan Dede Jumaenah. Peneliti menempuh pendidikan formalnya di SDN Bojong Rawalumbu X, SMPN 8 Bekasi, dan SMAN 15 Bekasi.

Pada tahun 2010, peneliti diterima sebagai mahasiswa Program Studi Pendidikan Teknik Elektronika, Jurusan Teknik Elektro, Fakultas Teknik, Universitas Negeri Jakarta melalui jalur SNMPTN. Dalam menyelesaikan studinya, peneliti mengadakan sebuah penelitian untuk pengerjaan skripsi dengan judul “*Vending machine* Minuman Kaleng Berbasis Arduino Mega 2560” sebagai syarat dalam mendapat gelar sarjana pendidikan.