

Rancang Bangun Sistem Sortir Pada Kotak Penyimpanan Uang (*MoneyBox Plus*) Berbasis Mikrokontroler Arduino

Muh. Devangga Hamdan Maulana¹, Firman Hadi Sukma Pratama²
Program Studi Informatika, Fakultas Teknik, Universitas Wijaya Kusuma Surabaya
devangga19@gmail.com¹, firmanpratama@uwks.ac.id²

Abstrak

Meski sudah banyak media penyimpanan uang yang praktis, masih ada beberapa orang lebih memilih menyimpan uang nya secara pribadi, yang dikarenakan tidak semua Bank memiliki banyak cabang atau mesin setor tunai yang banyak, serta adanya ketidak seimbangan dalam jarak tempuh menuju alat setor tunai yang jauh. Perancangan alat *MoneyBox Plus* dengan sistem sortir uang yang bisa memisahkan uang kertas dengan nominal Rp 50.000,00 dan Rp 100.000,00. Ini adalah penerapan mata *Internet Of Things* dalam program studi Informatika, yang dikembangkan dengan menggunakan metode *prototype* yang dimana *MoneyBox Plus* ini menggunakan beberapa komponen seperti *Motor Servo* dan mikrokontroler arduino. Hasil yang dituju dalam penelitian ini adalah bisa mensortir uang nominal pecahan uang Rp.50.000 dan Rp.100.000 pada tempat yang telah disediakan didalam *MoneyBox Plus*.

Kata Kunci : Sistem Sortir Uang, *Internet Of Things*, *Motor Servo*, Mikrokontroler Arduino, *MoneyBox Plus*

Abstract

The MoneyBox Plus is an innovative money storage device designed to cater to individuals who prefer to privately store their money due to limited access to traditional banking services. This project aims to develop a money sorting system using the Internet of Things (IoT) technology to separate banknotes of denominations Rp 50,000.00 and Rp 100,000.00. The implementation of a prototype using components such as Servo Motors and the Arduino microcontroller is utilized to achieve this goal. The primary objective of this research is to successfully sort and segregate the banknotes into their respective compartments within the MoneyBox Plus.

Keywords: Money sorting system, Internet of Things, Servo Motors, Arduino microcontroller, MoneyBox Plus.

I. PENDAHULUAN

Pada saat ini alat setor tunai terhitung sedikit daripada alat pengambil uang yang disediakan oleh Bank, dan kurangnya minat ketika hanya menyetorkan uang yang nominalnya sedikit dikarenakan tidak seimbang dengan jarak yang harus ditempuh menuju tempat yang memiliki mesin atau setor tunai sehingga lebih memilih untuk menyimpan uangnya secara pribadi dirumah.

Dari permasalahan tersebut maka dibuatnya alat yang bisa membantu dalam penyimpanan pribadi, alat ini dinamai *MoneyBox Plus* dan alat ini menggunakan *mikrokontroler Arduino* yang berfungsi untuk memproses *output* sesuai dengan yang di-*input*-kan.

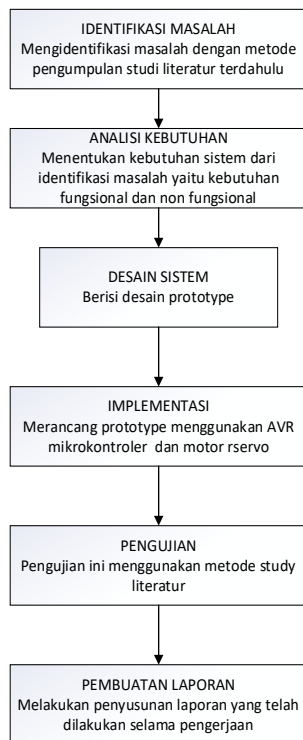
Pembuatan alat *MoneyBox Plus* dan alat ini sangat memerlukan sistem sortir uang pada kotak penyimpanan uang (*Money Box Plus*) berdasarkan inputan uang yang masuk agar dapat memisahkan uang pecahan Rp 50.000,00 dan Rp 100.000,00 di dalam *MoneyBox Plus* dan agar mempermudah untuk pengambilan uang pada *MoneyBox Plus*.

II. METODE

2.1. Tahapan Penelitian

2.1.1. Alur Penelitian

Penelitian Sistem Sortir Uang Pada Kotak Penyimpanan Uang (*MoneyBox Plus*) Berbasis Mikrokontroler Arduino dikerjakan menggunakan metode *prototype*. Alur penelitian ini digunakan sebagai acuan atau pedoman dalam proses pengerjaan penelitian yang agar dapat berjalan secara sistematis, efektif, dan diselesaikan tepat pada waktunya. Adapun tahapan tahapan sebagai berikut:



Gambar 2. 1 Diagram Alur

2.1.2. Identifikasi Masalah

Pada penelitian ini dibutuhkan cara dan Langkah dalam penyelesaian masalah merancang sistem sortir uang pada kotak penyimpanan uang (*MoneyBox Plus*) berbasis mikrokontroler Arduino. Penelitian ini mengimplementasikan Bahasa Pemrograman C sebagai dasar penelitian serta penelitian ini akan mendapatkan data yang akurat dari beberapa metode, diantara lain :

1. Studi Literatur

Studi literatur merupakan upaya mencari dan mempelajari berbagai sumber tulisan seperti jurnal, buku, laporan penelitian, situs internet dan berbagai artikel yang memiliki tema yang sama atau serupa dari sistem sortir uang pada kotak penyimpanan uang (*MoneyBox Plus*) berbasis mikrokontroler arduino. Studi literatur ini menjadi bagian penting untuk memperbanyak teori penelitian yang akan di uji

2. Percobaan Langsung

Percobaan pada sistem sortir uang pada kotak penyimpanan uang (*MoneyBox Plus*) berbasis mikrokontroler arduino digunakan untuk mengetahui apakah ada kendala dan masalah dalam perancangan *prototype* sehingga ada langkah perbaikan agar sistem berfungsi sesuai dengan tujuan yang diharapkan.

2.1.3. Analisis Kebutuhan

Setelah melakukan identifikasi masalah didapatkan data dan informasi yang berhubungan dengan sistem sortir uang pada penyimpanan uang (*MoneyBox Plus*) berbasis mikrokontroler Arduino dan akan dibangun sebuah *prototypenya*. Maka analisis kebutuhan akan dipecah menjadi 2 bagian yaitu Analisis Kebutuhan Fungsional dan Non Fungsional.

2.1.3.1. Analisis Kebutuhan Fungsional

Analisa kebutuhan fungsional adalah menjelaskan mengenai layanan atau fungsi apa saja yang disediakan oleh sistem untuk memudahkan user dalam menggunakannya. Di dalam kebutuhan fungsional ini terdiri dari input, proses, output, database, dan kebutuhan user. Kebutuhan fungsional dapat diartikan sebagai fitur apa saja yang dimiliki sistem atau kebutuhan yang berisi tentang proses-proses apa saja yang nantinya dilakukan oleh sistem. Adapun proses yang ada pada sistem ini nanti yaitu:

1. Pada perancangan sistem sortir uang pada kotak penyimpanan uang (*MoneyBox Plus*) berbasis mikrokontroler Arduino ini memiliki fitur yang bisa mensortir uang pada pecahan Rp.50.000 dan Rp.100.000 ke kotak yang sudah disediakan.

2.1.3.2. Analisis Kebutuhan Non Fungsional

Kebutuhan non-fungsional merupakan kebutuhan diluar kebutuhan fungsional yaitu meliputi kebutuhan akan spesifikasi *hardware* dan kebutuhan *software* yang sangat mempengaruhi beroperasinya mesin. Adapun kebutuhan non-fungsional tersebut meliputi :

1. Hardware :

Spesifikasi *hardware* minimum yang dibutuhkan pada perancangan sensor deteksi warna pada Kotak Penyimpanan Uang (*MoneyBox Plus*) berbasis mikrokontroler arduino adalah

1. PC dengan spesifikasi minimal :
 - Processor : 1.5 GHz Dual Core
 - Memory (RAM) : 2 GB DDR3
 - Harddisk : 120 GB
 - Monitor : resolusi Layar (1366 x 768)
 - Keyboard : QWERT (86 keys)
2. Sensor TCS34725
3. LCD
4. Kabel
5. Uang Rp.50.000 dan Rp.100.000

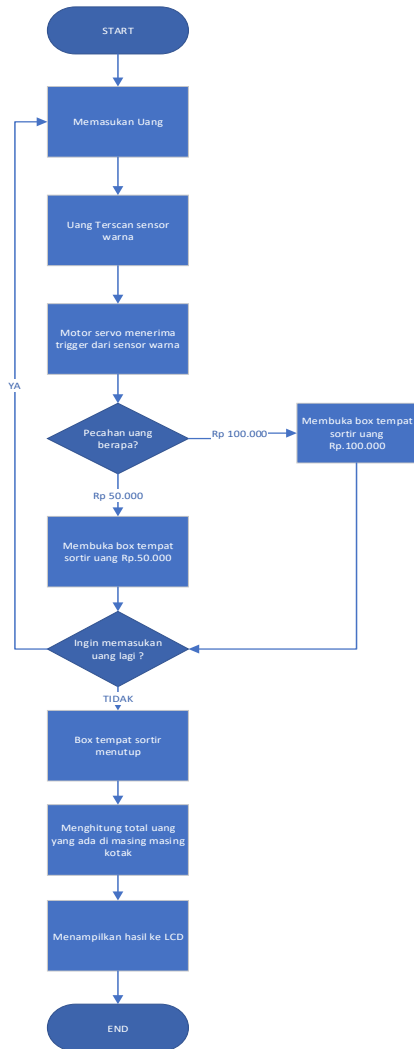
2. Software :

Spesifikasi minimal *software* minimum yang dibutuhkan pada perancangan sensor deteksi warna pada Kotak Penyimpanan Uang (*MoneyBox Plus*) berbasis mikrokontroler arduino adalah

1. Sistem Operasi :Windows 7
2. Aplikasi :Arduino IDE
3. Bahasa Pemrograman :Bahasa C

2.1.4. Flowchart

Adapun Flowchart yang digunakan pada sistem ini nanti yaitu:



Gambar 2. 2 Flowchart

2.1.5. Modeling Quick Design

Tahapan ini akan menggambarkan desain rancangan rangkaian sistem sortir uang pada kotak penyimpanan uang (MoneyBox Plus) secara kasar atau gambaran umum, adapun Modeling desain cepat yang akan digunakan sebagai berikut:



Gambar 2. 3 Modeling Quick Design

2.1.6. User Interface Design

Pada tahap user interface design atau UI Design ini akan menggambarkan desain rancangan dari antarmuka pengguna yang ditampilkan pada TFT LCD Display Shield dan digunakan untuk memilih Menu yang terdapat di Sistem Keamanan Pada Kotak Penyimpanan Uang (MoneyBox Plus) Berbasis Mikrokontroler Arduino. Adapun desain rancangan dari antarmuka pengguna yang akan digunakan adalah sebagai berikut.

2.1.6.1. Design UI Menu Utama

Rancangan Desain UI Menu Utama ini digunakan sebagai tampilan awal pada Kotak Penyimpanan Uang (MoneyBox Plus), dan di tampilan ini pengguna dapat memilih beberapa Menu yang tersedia pada sistem. Diantaranya yang termasuk ke dalam Sistem Sortir ini adalah Menu Akses Masuk, Menu Ubah PIN, dan Menu Lupa PIN.



Gambar 2. 4 Design UI Menu

2.1.6.2. Desain UI Menu Cek Jumlah

Rancangan Desain UI Menu cek jumlah ini digunakan sebagai tampilan atau layout yang terdapat instruksi "Masukkan PIN", selain itu juga berisi angka untuk memasukkan kode pengaman dan "OK" untuk konfirmasi kode pengaman, beserta "x" untuk cancel atau kembali ke Menu tampilan awal. Pada Menu Akses Masuk untuk inputan kode pengaman disembunyikan.

Masukan Pin		
1	2	3
4	5	6
7	8	9

1	2	3
4	5	6
7	8	9

Gambar 2. 5 Rancangan UI Menu masukan PIN

Saat melakukan input kode pengaman pada Menu Akses Masuk dan kode pengaman tersebut cocok atau benar dengan PIN yang sudah tersimpan ke dalam sistem, maka tampilan akan berubah menjadi seperti gambar berikut :

Total Uang Di Money Box Plus	
Jumlah Uang	
Reset	
<	

Gambar 2. 6 Rancangan UI Cek Jumlah Uang

2.1.7. Sketching Proyotype

Setelah membuat tahapan perencanaan adalah tahap mengubah modeling quick design ke *sketching prototype* untuk mempermudah pembuatan *prototype*, berikut adalah hasil dari *sketching protoype* :



Gambar 2. 7 Sketching Prototype

III. HASIL DAN PEMBAHASAN

3.1. Pengujian Sistem

Pengujian sistem disini berdasarkan rancangan desain yang telah dibuat dan untuk mengetahui apakah sistem yang telah dibuat dapat berfungsi dengan sesuai yang diharapkan atau tidak. Berikut adalah uji coba sistem dari pengerjaan sistem berdasarkan rancangan desain yang telah dibuat. Adapun implementasinya pada Kotak Penyimpanan Uang (MoneyBos Plus) sebagai berikut :

3.1.1. Menu Utama

Uji coba pada menu utama ini merupakan akses yang akan digunakan sebagai tampilan awal yang nantinya bisa menjadi jembatan kepada fitur scan uang yang ada pada Kotak Penyimpanan Uang (MoneyBox Plus).



Gambar 3. 1 Menu Utama MoneyBOX Plus

3.1.2. Menu Cek Jumlah

Menu Cek Jumlah merupakan akses yang di gunakan untuk melihat jumlah uang pada Kotak Penyimpanan Uang (MoneyBox Plus) dan hanya orang berhak saja yang memiliki akses. Adapun Pengujian Menu Akses Masuk ini adalah sebagai berikut.

1. Setelah masuk ke Menu Akses Masuk, masukkan PIN atau kode pengaman yang sesuai dengan PIN atau kode pengaman pada sistem untuk masuk ke dalam akses Kotak Penyimpanan Uang (MoneyBox Plus).



Gambar 3. 2 Gambar UI Cek Jumlah

2. Jika PIN atau kode pengaman yang dimasukkan sesuai dengan PIN atau kode pengaman pada sistem, maka akan muncul jumlah uang yang telah tersimpan pada Kotak Penyimpanan Uang (MoneyBox Plus).



Gambar 3. 3 UI menampilkan jumlah uang

3. Sedangkan jika PIN atau kode pengaman yang dimasukkan tidak sesuai dengan PIN atau kode pengaman pada sistem, maka akses ditolak dan Kotak Penyimpanan Uang (MoneyBox Plus) tetap terkunci.



Gambar 3. 4 Akses Cek Jumlah Ditolak

3.1.3. Penerapan Cara Kerja Sistem Sortir

Penerapan cara kerja deteksi warna pada Kotak Penyimpanan Uang (MoneyBox Plus) sebagai berikut :

1. Sambungkan alat MoneyBox Plus pada Power Supply, yang dimana alat yang digunakan sebagai power supply ini menggunakan kabel data dari USB.
2. Setelah alat terhubung dengan power supply maka arduino yang bekerja sebagai otak mulai bekerja untuk menjalankan mesin pada Kotak Penyimpanan Uang (MoneyBox Plus).
3. Selanjutnya, menu utama dari Kotak Penyimpanan Uang ini akan menyala.
4. Tekan tombol pada layar dibagian Scan Uang, dan nanti akan pindah ke bagian Scan Uang.
5. Masukkan uang pecahan Rp.50.000 atau Rp.100.000 yang mau ditaruh ke Kotak Penyimpanan Uang (MoneyBox Plus), batas memasukkan uang dalam satu fase adalah 5 kali memasukkan uang setelah itu Kotak Penyimpanan Uang (MoneyBox Plus) akan menampilkan total uang yang sudah discan melalui scan warna.
6. Setelah memasukan uang pecahan Rp. 50.000 atau Rp.100.000 Motor Servo yang ada pada Kotak Penyimpanan uang (MoneyBox Plus) akan ke-Trigger.
7. Jika Motor Servo sudah ke-Trigger maka akan bergerak untuk membuka kotak pemisah yang ada di dalam Kotak Penyimpanan uang (MoneyBox Plus).
8. Setelah itu akan terbuka masing-masing kotak yang ada pada Kotak Penyimpanan uang (MoneyBox Plus) yang dimana akan mensortir uang pecahan Rp.50.000 atau Rp.100.000 sesuai kotak yang telah tersedia di dalam Kotak Penyimpanan uang (MoneyBox Plus).

3.2. Hasil Penelitian

Berdasarkan pengujian sistem yang dilakukan saat penelitian terhadap kotak Penyimpanan Uang (MoneyBos Plus) memperoleh hasil sebagai berikut :

3.2.1. Layout Kode Menu Utama

```

324 void Menu_display()
325 {
326     tft.setTextSize(3);
327     tft.setTextColor(BLACK);
328     tft.setCursor(40, 37);
329     tft.print("MoneyBox");
330
331     tft.setTextSize(3);
332     tft.setTextColor(BLACK);
333     tft.setCursor(185, 22);
334     tft.print("+");
335
336     DrawButtonScanUang();
337     DrawButtonCekJumlah();
338     DrawButtonAksesMasuk();
339     DrawButtonResetPIN();
340     DrawButtonLupaPIN();
341     DrawButtonClear();
342 }
343

```

Gambar 3. 5 Source Code Menu Utama

Pada gambar diatas, penjelasan mengenai kode dalam fungsi "Menu Display" pada program Arduino adalah sebagai berikut:

1. ``tft.setTextSize(3);``: Mengatur ukuran teks yang akan ditampilkan pada layar menjadi 3 kali ukuran default.
2. ``tft.setTextColor(BLACK);``: Mengatur warna teks menjadi hitam.
3. ``tft.setCursor(40, 37);``: Mengatur posisi kursor teks pada layar dengan koordinat (40, 37). Koordinat ini menentukan letak awal teks yang akan ditampilkan.
4. ``tft.print("MoneyBox");``: Mencetak teks "MoneyBox" pada layar pada posisi yang telah ditentukan sebelumnya.
5. ``tft.setTextSize(3);``: Mengatur ukuran teks kembali menjadi 3 kali ukuran default.
6. ``tft.setTextColor(BLACK);``: Mengatur warna teks kembali menjadi hitam.
7. ``tft.setCursor(185, 22);``: Mengatur posisi kursor teks pada layar dengan koordinat (185, 22).
8. ``tft.print("+");``: Mencetak tanda "+" pada layar pada posisi yang telah ditentukan sebelumnya.
9. Selanjutnya, terdapat pemanggilan beberapa fungsi lain, seperti ``DrawButtonScanUang()`,
`DrawButtonCekJumlah()`,
`DrawButtonAksesMasuk()`,
`DrawButtonResetPIN()`,
`DrawButtonLupaPIN()`,
`DrawButtonClear()`. Fungsi-fungsi ini bertanggung jawab untuk menggambar dan menampilkan tombol-tombol pada layar. Dengan menjalankan fungsi "Menu Display", layar akan menampilkan teks "MoneyBox" dan beberapa tombol yang tergambar sesuai dengan pemanggilan fungsi-fungsi yang ada di dalamnya. Fungsi ini bertanggung jawab untuk menampilkan tampilan menu utama pada layar TFT (Thin-Film Transistor) display pada Arduino.`

3.2.2. Layout Kode Menu Cek Jumlah

```
void DrawButtonCekJumlah()
{
  tft.fillRoundRect(123, 95, 95, 70, 10, BLACKM);
  tft.fillRoundRect(125, 97, 91, 66, 10, BLUE);
  tft.setTextSize(2);
  tft.setTextColor(WHITE);
  tft.setCursor(153, 110);
  tft.print("CEK");
  tft.setCursor(135, 135);
  tft.print("JUMLAH");
}
```

Gambar 3. 6 Source Code Menu Cek Jumlah Fungsi 1

Pada gambar diatas merupakan fungsi pertama, `DrawButtonCekJumlah()`, digunakan untuk menggambar tombol dengan teks "CEK

JUMLAH". Berikut adalah langkah-langkah yang diambil dalam fungsi tersebut :

1. ``tft.fillRoundRect(123, 95, 95, 70, 10, BLACKM);``: Fungsi ini menggambar sebuah persegi panjang dengan sudut yang dibulatkan pada posisi (123, 95) dengan lebar 95 piksel dan tinggi 70 piksel. Parameter terakhir, ``BLACKM``, kemungkinan merujuk pada warna hitam dalam format yang digunakan oleh library yang digunakan dalam proyek ini.
2. ``tft.fillRoundRect(125, 97, 91, 66, 10, BLUE);``: Fungsi ini menggambar persegi panjang dengan sudut yang dibulatkan pada posisi (125, 97) dengan lebar 91 piksel dan tinggi 66 piksel. Perbedaannya adalah warna yang digunakan kali ini adalah biru (BLUE).
3. ``tft.setTextSize(2);``: Fungsi ini mengatur ukuran teks pada layar menjadi 2.
4. ``tft.setTextColor(WHITE);``: Fungsi ini mengatur warna teks menjadi putih (WHITE).
5. ``tft.setCursor(153, 110);``: Fungsi ini menentukan posisi kursor untuk penulisan teks berikutnya. Dalam hal ini, kursor ditempatkan pada posisi (153, 110).
6. ``tft.print("CEK");``: Fungsi ini mencetak teks "CEK" pada posisi kursor yang telah ditentukan sebelumnya.
7. ``tft.setCursor(135, 135);``: Fungsi ini menentukan posisi kursor untuk penulisan teks selanjutnya. Dalam hal ini, kursor ditempatkan pada posisi (135, 135).
8. ``tft.print("JUMLAH");``: Fungsi ini mencetak teks "JUMLAH" pada posisi kursor yang baru ditentukan.

```
373 void DrawButtonCekJumlahPress()
374 {
375   tft.fillRoundRect(123, 95, 95, 70, 10, BLACKM);
376 }
```

Gambar 3. 7 Source Code Menu Cek Jumlah Fungsi 2

Pada gambar di atas merupakan Fungsi ``DrawButtonCekJumlahPress()`. Berikut adalah langkah-langkah yang diambil dalam fungsi tersebut :

1. ``tft.fillRoundRect(123, 95, 95, 70, 10, BLACKM);``: Fungsi ini menggambar persegi panjang dengan sudut yang dibulatkan pada posisi (123, 95) dengan lebar 95 piksel dan tinggi 70 piksel. Perbedaannya adalah kali ini persegi

panjang diisi dengan warna hitam (BLACKM), menciptakan efek tekanan pada tombol.

```

1451 void writeIntIntoEEPROM(int address, int number)
1452 {
1453     EEPROM.write(address, number >> 8);
1454     EEPROM.write(address + 1, number & 0xFF);
1455 }

```

Gambar 3. 8 Source Code Menu Cek Jumlah Fungsi 3

Pada gambar diatas merupakan Fungsi `writeIntIntoEEPROM(int address, int number)` Fungsi ini digunakan untuk menulis sebuah bilangan bulat (*integer*) ke dalam EEPROM (*Electrically Erasable Programmable Read-Only Memory*). EEPROM adalah jenis memori yang dapat menyimpan data bahkan ketika daya listrik terputus. Berikut adalah langkah-langkah yang diambil dalam fungsi tersebut :

1. `EEPROM.write(address, number >> 8);`: Pada baris ini, kita menggunakan fungsi `EEPROM.write()` untuk menulis data ke alamat yang ditentukan dalam EEPROM. Parameter pertama, `address`, adalah alamat memori EEPROM di mana kita ingin menyimpan bilangan tersebut. Parameter kedua, `number >> 8`, adalah bilangan yang akan ditulis ke EEPROM setelah digeser 8 bit ke kanan. Hal ini dilakukan untuk mendapatkan 8 bit paling signifikan (*most significant bits/MSB*) dari bilangan `number`.
2. `EEPROM.write(address + 1, number & 0xFF);`: Pada baris ini, kita menggunakan fungsi `EEPROM.write()` sekali lagi untuk menulis data ke alamat yang berikutnya dalam EEPROM. Alamat yang digunakan adalah `address + 1`, untuk menulis 8 bit paling tidak signifikan (*least significant bits/LSB*) dari bilangan `number`. `number & 0xFF` mengambil 8 bit terakhir dari `number` dengan menggunakan operasi bitwise AND dengan `0xFF`.

```

1457 int readIntFromEEPROM(int address)
1458 {
1459     return (EEPROM.read(address) << 8) + EEPROM.read(address + 1);
1460 }
1461

```

Gambar 3. 9 Source Code Menu Cek Jumlah Fungsi 4

Pada gambar diatas merupakan fungsi `readIntFromEEPROM(int address)` Fungsi ini digunakan untuk membaca sebuah bilangan bulat (*integer*) dari EEPROM. Berikut adalah langkah-langkah yang diambil dalam fungsi tersebut :

1. `(EEPROM.read(address) << 8) + EEPROM.read(address + 1);`: Baris ini membaca data dari EEPROM dengan menggunakan fungsi `EEPROM.read()`. Kita membaca byte pertama dari alamat yang ditentukan, `EEPROM.read(address)`, kemudian menggeser (*shift*) nilai byte tersebut ke kiri (*left shift*) sebanyak 8 bit menggunakan operator bitwise shift (`<< 8`).
2. Setelah itu, kita menambahkan hasil dari pergeseran bit tersebut dengan byte kedua yang dibaca dari alamat berikutnya, `EEPROM.read(address + 1)`. Hal ini dilakukan untuk menggabungkan nilai byte pertama (*MSB*) dengan byte kedua (*LSB*) dan menghasilkan bilangan bulat yang akan dikembalikan oleh fungsi.

```

1462 void SimpanTotalUang()
1463 {
1464     jumlahUang = readIntFromEEPROM(23);
1465     jumlahUang = jumlahUang + totalSementara;
1466     writeIntIntoEEPROM(23, jumlahUang);
1467 }

```

Gambar 3. 10 Source Code Menu Cek Jumlah Fungsi 5

Pada gambar diatas merupakan fungsi `SimpanTotalUang()` Fungsi ini digunakan untuk menyimpan total uang ke dalam EEPROM. Berikut adalah langkah-langkah yang diambil dalam fungsi tersebut :

1. `jumlahUang = readIntFromEEPROM(23);`: Pada baris ini, kita menggunakan fungsi `readIntFromEEPROM()` untuk membaca nilai total uang yang telah disimpan sebelumnya dari alamat EEPROM 23. Nilai yang dibaca kemudian disimpan dalam variabel `jumlahUang`.
2. `jumlahUang = jumlahUang + totalSementara;`: Setelah itu, kita menambahkan nilai `totalSementara` ke dalam `jumlahUang`. Ini bertujuan untuk mengupdate nilai total uang dengan menambahkan nilai sementara yang baru.
3. `writeIntIntoEEPROM(23, jumlahUang);`: Pada baris ini, kita menggunakan fungsi `writeIntIntoEEPROM()` untuk menulis nilai yang telah diubah ke dalam alamat EEPROM 23.

```

1469 void TampilanResetTotalUang()
1470 {
1471     Serial.println("Total Uang Telah Direset");
1472     tft.fillScreen(WHITE);
1473     tft.setTextSize(3);
1474     tft.setTextColor(JADE);
1475     tft.setCursor(28, 124);
1476     tft.println("Total Uang");
1477     tft.setCursor(3, 164);
1478     tft.println("Telah Direset");
1479 }
    
```

Gambar 3. 11 Source Code Menu Cek Jumlah fungsi 6

Pada gambar diatas merupakan fungsi TampilanResetTotalUang(). Fungsi ini bertujuan untuk menampilkan pesan bahwa total uang telah direset dan melakukan beberapa pengaturan tampilan pada layar TFT (Thin-Film Transistor). Berikut adalah langkah-langkah yang diambil dalam fungsi tersebut :

1. Serial.println("Total Uang Telah Direset");: Pada baris ini, pesan "Total Uang Telah Direset" akan dicetak ke output Serial Monitor. Ini berguna jika Anda menggunakan koneksi serial untuk melihat pesan ini dalam lingkungan pemrograman Arduino atau sejenisnya.
2. tft.fillScreen(WHITE);: Baris ini akan mengisi seluruh layar TFT dengan warna putih. tft adalah objek yang digunakan untuk mengendalikan tampilan pada layar TFT.
3. tft.setTextSize(3);: Pada baris ini, ukuran teks pada layar TFT diatur menjadi 3. Ini menentukan ukuran teks yang akan digunakan untuk teks selanjutnya yang ditampilkan pada layar.
4. tft.setTextColor(JADE);: Baris ini mengatur warna teks pada layar TFT menjadi warna "JADE". "JADE" mungkin merupakan konstanta yang didefinisikan sebelumnya dalam program atau kode Anda.
5. tft.setCursor(28, 124);: Pada baris ini, posisi kursor teks pada layar TFT diatur ke koordinat (28, 124). Ini menentukan posisi awal di mana teks akan ditampilkan pada layar.
6. tft.println("Total Uang");: Baris ini akan mencetak teks "Total Uang" pada layar TFT sesuai dengan posisi kursor yang telah diatur sebelumnya.
7. tft.setCursor(3, 164);: Pada baris ini, posisi kursor teks pada layar TFT diatur ke koordinat (3, 164). Ini menentukan posisi awal di mana teks selanjutnya akan ditampilkan pada layar.
8. tft.println("Telah Direset");: Baris ini akan mencetak teks "Telah Direset" pada layar TFT

sesuai dengan posisi kursor yang telah diatur sebelumnya.

Dengan demikian, fungsi TampilanResetTotalUang() akan menampilkan pesan "Total Uang Telah Direset" pada output Serial Monitor (jika ada) dan melakukan pengaturan tampilan pada layar TFT untuk menampilkan teks "Total Uang" dan "Telah Direset".

3.2.3. Layout Kode Motor Servo

```

83 int position1=100, position2=140;
    
```

Gambar 3. 12 Source Code Motor Servo fungsi 1

Pada gambar diatas merupakan Source Code yang berfungsi untuk mendefinisikan dua variabel bertipe int, yaitu position1 dan position2, dan memberikan nilai awal masing-masing variabel tersebut. Berikut adalah langkah-langkah yang diambil dalam fungsi tersebut :

1. int position1 = 100;; Baris ini mendefinisikan variabel *position1* dengan tipe data int (integer) dan memberikan nilai awal 100 ke variabel tersebut. Variabel *position1* kemungkinan digunakan untuk menyimpan posisi yang akan diberikan ke servo yang terhubung ke servo_28.
2. int position2 = 140;; Baris ini mendefinisikan variabel *position2* dengan tipe data int (integer) dan memberikan nilai awal 140 ke variabel tersebut. Variabel *position2* kemungkinan digunakan untuk menyimpan posisi yang akan diberikan ke servo yang terhubung ke servo_30.

Dengan memberikan nilai awal ini dapat diatur posisi awal dari servo motor ketika program dijalankan, dan dapat mengubah nilai position1 dan position2 sesuai dengan kebutuhan untuk mengatur posisi servo motor yang diinginkan pada awal program.

```

76 Servo servo;
77 Servo servo_28;
78 Servo servo_30;
    
```

Gambar 3. 13 Source Code Motor Servo fungsi 2

Pada gambar diatas mendefinisikan tiga objek Servo dengan nama servo, servo_28, dan servo_30. Objek-objek ini digunakan untuk mengendalikan servo motor pada mikrokontroler atau papan Arduino. Berikut adalah langkah-langkah yang diambil dalam fungsi tersebut :

1. Servo servo;; Baris ini mendefinisikan objek servo dengan tipe data Servo. Objek ini

digunakan untuk mengendalikan sebuah servo motor.

2. Servo servo_28;: Baris ini mendefinisikan objek servo_28 dengan tipe data Servo. Objek ini juga digunakan untuk mengendalikan sebuah servo motor.
3. Servo servo_30;: Baris ini mendefinisikan objek servo_30 dengan tipe data Servo. Objek ini juga digunakan untuk mengendalikan sebuah servo motor.

Dengan mendefinisikan objek-objek Servo ini dapat menggunakan objek-objek tersebut untuk menghubungkan servo motor ke pin tertentu pada mikrokontroler atau papan Arduino, serta menggerakkan servo ke posisi yang diinginkan menggunakan fungsi-fungsi yang disediakan oleh library Servo. Misalnya, Anda dapat menggunakan fungsi `attach()` untuk menghubungkan servo ke pin tertentu, dan fungsi `write()` untuk menggerakkan servo ke posisi tertentu.

```
124 servo_28.attach(28);
125 servo_30.attach(30);
126
127 servo_28.write(position1);
128 servo_30.write(position2);
```

Gambar 3. 14 Source Code Motor Servo fungsi 3

Pada gambar diatas merupakan Source Code yang berfungsi untuk mengontrol dua servo motor. Mari kita jelaskan setiap barisnya. Berikut adalah langkah-langkah yang diambil dalam fungsi tersebut :

1. `servo_28.attach(28);`: Pada baris ini, servo dengan nama `servo_28` dihubungkan ke pin digital 28 pada mikrokontroler atau papan Arduino yang digunakan. Fungsi `attach()` digunakan untuk menghubungkan objek servo ke pin yang ditentukan.
2. `servo_30.attach(30);`: Pada baris ini, servo dengan nama `servo_30` dihubungkan ke pin digital 30 pada mikrokontroler atau papan Arduino yang digunakan. Fungsi `attach()` digunakan untuk menghubungkan objek servo ke pin yang ditentukan.
3. `servo_28.write(position1);`: Baris ini mengatur posisi servo `servo_28` sesuai dengan nilai yang ada dalam variabel `position1`. Fungsi `write()` digunakan untuk menggerakkan servo ke posisi tertentu yang diberikan dalam parameter.

4. `servo_30.write(position2);`: Baris ini mengatur posisi servo `servo_30` sesuai dengan nilai yang ada dalam variabel `position2`. Fungsi `write()` digunakan untuk menggerakkan servo ke posisi tertentu yang diberikan dalam parameter.

Dalam kode tersebut, servo motor dihubungkan ke pin digital 28 dan 30, dan kemudian digerakkan ke posisi yang telah ditentukan oleh variabel `position1` dan `position2`. Anda perlu memastikan bahwa pin-pin yang digunakan sesuai dengan koneksi fisik pada mikrokontroler atau papan Arduino yang Anda gunakan, dan juga memastikan bahwa variabel `position1` dan `position2` berisi nilai posisi yang *valid* untuk servo motor yang digunakan.

```
if (r>48 && r<70 && g>102 && g<117 && b>86 && b<110)
{
  tft.fillRoundRect(15, 100, 210, 80, 5, BLACKM);
  tft.fillRoundRect(17, 102, 206, 76, 5, LIMAPULUH);

  tft.setCursor(40, 129);
  tft.setTextSize(3);
  tft.setTextColor(WHITE);
  tft.println("Rp.50.000");
  limapuluh = 50;

  servo_28.write(65);
  delay(3000);
  servo_28.write(position1);
}
```

Gambar 3. 15 Source Code Motor Servo fungsi 4

Pada gambar diatas merupakan Source Code adalah sebuah blok `if` yang melakukan pengecekan kondisi pada variabel `r`, `g`, dan `b`. Jika kondisi tersebut terpenuhi, maka blok kode di dalamnya akan dieksekusi. Berikut adalah langkah-langkah yang diambil dalam fungsi tersebut:

1. `if (r > 48 && r < 70 && g > 102 && g < 117 && b > 86 && b < 110)`: Baris ini adalah kondisi yang harus terpenuhi agar blok kode di dalamnya dieksekusi. Kondisi ini memeriksa apakah nilai variabel `r` berada di antara 48 dan 70, nilai variabel `g` berada di antara 102 dan 117, serta nilai variabel `b` berada di antara 86 dan 110.
2. `{`: Kurung kurawal membuka blok kode yang akan dieksekusi jika kondisi di atas terpenuhi.
3. `tft.fillRoundRect(15, 100, 210, 80, 5, BLACKM);`: Baris ini mengisi suatu area persegi panjang dengan sudut melengkung pada layar TFT menggunakan warna `BLACKM`. Area yang diisi dimulai dari koordinat (15, 100) dengan lebar 210 piksel dan tinggi 80 piksel.

4. `tft.fillRoundRect(17, 102, 206, 76, 5, LIMAPULUH);`: Baris ini mengisi suatu area persegi panjang dengan sudut melengkung pada layar TFT menggunakan warna LIMAPULUH. Area yang diisi dimulai dari koordinat (17, 102) dengan lebar 206 piksel dan tinggi 76 piksel.
5. `tft.setCursor(40, 129);`: Baris ini mengatur posisi kursor pada layar TFT ke koordinat (40, 129). Kursor ini akan digunakan untuk menampilkan teks selanjutnya pada layar.
6. `tft.setTextSize(3);`: Baris ini mengatur ukuran teks pada layar TFT menjadi 3.
7. `tft.setTextColor(WHITE);`: Baris ini mengatur warna teks pada layar TFT menjadi WHITE.
8. `tft.println("Rp.50.000");`: Baris ini mencetak teks "Rp.50.000" pada layar TFT sesuai dengan posisi kursor yang telah diatur sebelumnya.
9. `limapuluh = 50;`: Baris ini mengassign nilai 50 ke variabel limapuluh.
10. `servo_28.write(65);`: Baris ini menggerakkan servo yang terhubung ke servo_28 ke posisi 65. Fungsi `write()` digunakan untuk menggerakkan servo ke posisi tertentu yang diberikan dalam parameter.
11. `delay(3000);`: Baris ini menyebabkan program menghentikan eksekusi selama 3000 milidetik (3 detik).
12. `servo_28.write(position1);`: Baris ini menggerakkan servo yang terhubung ke servo_28 ke posisi yang ditentukan oleh variabel position1.

Dengan demikian, jika kondisi pada baris pertama terpenuhi, maka blok kode di dalamnya akan di eksekusi dan *Motor Servo* akan bergerak.

```

1123 else if (r1>150 && r1<175 && g1>166 && g1<190 && b1>135 && b1<177)
1124 {
1125     tft.fillRoundRect(15, 100, 210, 80, 5, BLACKM);
1126     tft.fillRoundRect(17, 102, 206, 76, 5, SERATUS);
1127
1128     tft.setCursor(31, 129);
1129     tft.setTextSize(3);
1130     tft.setTextColor(WHITE);
1131     tft.println("Rp.100.000");
1132     seratus = 100;
1133
1134
1135     servo_30.write(65);
1136     delay(3000);
1137     servo_30.write(position2);
1138 }
    
```

Gambar 3. 16 Source Code Motor Servo fungsi 5

Pada gambar diatas merupakan Source Code yang terdapat blok else if yang memeriksa kondisi lainnya. Berikut adalah langkah-langkah yang diambil dalam fungsi tersebut:

1. Kondisi else if: Ini adalah kondisi kedua yang dievaluasi jika kondisi pertama tidak terpenuhi. Dalam kondisi ini, kita memeriksa apakah nilai r1, g1, dan b1 berada dalam rentang tertentu.
2. Blok kode di dalam kondisi else if akan dieksekusi jika kondisi tersebut terpenuhi.
3. `tft.fillRoundRect()`: Fungsi ini mengisi suatu area persegi panjang dengan sudut melengkung pada layar TFT. Pada baris ini, kita mengisi area persegi panjang dengan sudut melengkung menggunakan warna BLACKM.
4. `tft.setCursor()`: Fungsi ini mengatur posisi kursor pada layar TFT menggunakan koordinat x dan y.
5. `tft.setTextSize()`: Fungsi ini mengatur ukuran teks pada layar TFT.
6. `tft.setTextColor()`: Fungsi ini mengatur warna teks pada layar TFT.
7. `tft.println()`: Fungsi ini mencetak teks pada layar TFT diikuti dengan karakter baru.
8. `seratus = 100;`: Nilai variabel seratus diatur menjadi 100.
9. `servo_30.write()`: Fungsi ini menggerakkan servo yang terhubung ke pin 30 ke posisi 65.
10. `delay()`: Fungsi ini menunda eksekusi program selama 3000 milidetik (3 detik).
11. `servo_30.write(position2);`: Setelah jeda 3 detik, servo yang terhubung ke pin 30 digerakkan ke posisi yang ditentukan oleh variabel position2.

Dengan demikian, jika kondisi pada baris kedua terpenuhi, maka blok kode di dalamnya akan di eksekusi dan *Motor Servo* akan bergerak.

```

1140 else
1141 {
1142     tft.fillRoundRect(15, 100, 210, 80, 5, BLACKM);
1143     tft.fillRoundRect(17, 102, 206, 76, 5, WHITE);
1144
1145     tft.setCursor(85, 129);
1146     tft.setTextSize(3);
1147     tft.setTextColor(BLACK);
1148     tft.println("Rp.0");
1149     nol = 0;
1150
1151
1152     servo.write(0);
1153
1154 }
    
```

Gambar 3. 17 Source Code Motor Servo fungsi 6

Pada gambar diatas merupakan Source Code terdapat blok else yang akan dieksekusi jika kedua kondisi sebelumnya tidak terpenuhi. Berikut adalah langkah-langkah yang diambil dalam fungsi tersebut:

1. else: Ini adalah blok yang akan dieksekusi jika kedua kondisi sebelumnya tidak terpenuhi.
2. Blok kode di dalam else akan dieksekusi jika kondisi tersebut tidak terpenuhi.
3. `tft.fillRoundRect()`: Fungsi ini mengisi suatu area persegi panjang dengan sudut melengkung pada layar TFT. Pada baris ini, kita mengisi area persegi panjang dengan sudut melengkung menggunakan warna BLACKM.
4. `tft.setCursor()`: Fungsi ini mengatur posisi kursor pada layar TFT menggunakan koordinat x dan y.
5. `tft.setTextSize()`: Fungsi ini mengatur ukuran teks pada layar TFT.
6. `tft.setTextColor()`: Fungsi ini mengatur warna teks pada layar TFT.
7. `tft.println()`: Fungsi ini mencetak teks pada layar TFT diikuti dengan karakter baru.
8. `nol = 0;`: Nilai variabel nol diatur menjadi 0.
9. `servo.write(0);`: Fungsi ini menggerakkan servo ke posisi 0.

Dengan demikian, jika kondisi pada baris ketiga terpenuhi, maka blok kode di dalamnya akan eksekusi dan *Motor Servo* tidak akan bergerak.

3.3. Hasil Implementasi

Hasil implementasi disini merupakan hasil dari prototype atau perancangan Kotak Penyimpanan Uang (*MoneyBox Plus*) dengan berbagai sistem. Salah satunya sistem sortir ini didapatkan hasil yang di harapkan. *Motor Servo* mampu menggerakkan kotak yang terdapat pada *MoneyBox Plus* dengan akurat dan memberikan respons sesuai kondisi yang telah di tentukan untuk pecahan uang Rp.50.000 dan Rp.100.000. Melalui penggunaan kondisional if-else, sistem dapat mengidentifikasi dengan tepat pecahan uang yang dimasukkan ke dalam *MoneyBox Plus*.

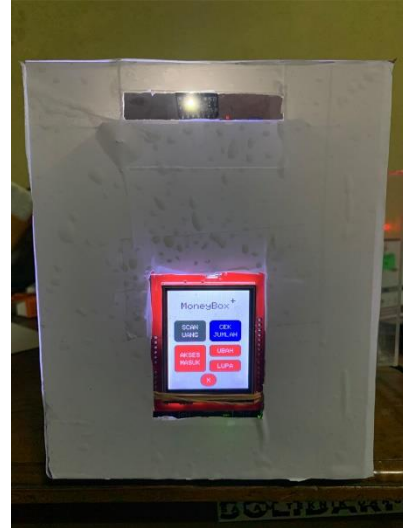
Selain itu, hasil uji coba juga memvalidasi kinerja layar TFT yang digunakan untuk menampilkan informasi mengenai pecahan uang yang terdeteksi. Informasi yang ditampilkan dengan jelas dan akurat, memudahkan pengguna dalam memonitor jumlah uang yang telah dimasukkan ke dalam kotak penyimpanan.

Dengan hasil uji coba yang positif ini, dapat disimpulkan bahwa implementasi sensor deteksi warna pada *MoneyBox Plus* telah berhasil dan sesuai dengan tujuan penelitian. Hasil ini memberikan bukti bahwa sistem dapat digunakan

sebagai alat bantu penyimpanan uang yang efektif dan efisien.

3.3.1. Penggunaan Layar LCD

Pada Kotak Penyimpanan Uang (*MoneyBox Plus*) ini, Modul TFT LCD *Display Shield* digunakan untuk menampilkan informasi yang tersedia pada sistem, dan juga sebagai kontrol utama pada sistem. Modul ini dapat dihubungkan secara langsung pada board *Arduino Mega 2560*.



Gambar 3. 18 Penggunaan Layar LCD

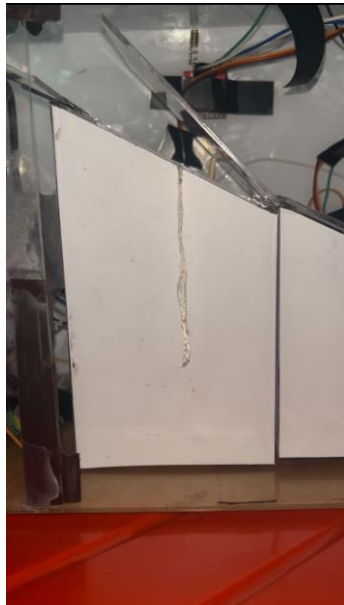
3.3.2. Penggunaan Motor Servo

Sistem sortir pada Kotak Penyimpanan Uang (*MoneyBox Plus*) ini menggunakan *Motor Servo* yang berfungsi sebagai penggerak kotak yang terdapat didalam Kotak Penyimpanan Uang (*MoneyBox Plus*). *Motor Servo* ini dapat bergerak secara otomatis untuk dapat memisahkan uang pecahan Rp. 50.000 dan Rp. 100.000 .



Gambar 3. 19 Kotak pecahan uang RP. 100.000 terbuka

Pada gambar diatas memperlihatkan ketika uang pecahan RP. 100.000 terdeteksi maka kotak untuk uang pecahan RP. 100.000 terbuka secara otomatis.



Gambar 3. 20 Kotak pecahan uang RP. 50.000 terbuka

Pada gambar diatas memperlihatkan ketika uang pecahan RP. 50.000 terdeteksi maka kotak untuk uang pecahan RP. 50.000 terbuka secara otomatis.

IV. PENUTUP

4.1. Kesimpulan

Dari hasil pengujian Rancang Bangun Sistem Keamanan Pada Kotak Penyimpanan Uang (*MoneyBox Plus*) Berbasis Mikrokontroler Arduino dapat disimpulkan :

1. Sistem sortir pada Kotak Penyimpanan Uang (*MoneyBox Plus*) berfungsi dengan baik dan sesuai dengan kondisi *Motor Servo* saat menerima proses dari sistem.
2. *Layout* pada menu cek jumlah uang dapat bekerja dengan baik dan dapat menampilkan jumlah uang yang telah masuk dengan akurat.
3. Sistem sortir pada Kotak Penyimpanan Uang (*MoneyBox Plus*) dapat memudahkan untuk pengambilan uang pada *Money Box Plus* dikarenakan adanya kotak pemisah untuk uang pecahan Rp. 50.000 dan Rp. 100.000

4.2. Saran

Dalam laporan tugas akhir yang telah dibuat oleh penulis, maka penulis mencoba memberikan saran untuk pengembangan alat atau sistem ini kedepannya :

1. Pengembangan Fitur Tambahan: *MoneyBox Plus* dapat terus dikembangkan dengan

menambahkan fitur-fitur tambahan yang dapat meningkatkan fungsionalitas dan kenyamanan pengguna. Misalnya, integrasi dengan sistem keamanan tambahan seperti pengenalan wajah atau sidik jari untuk mengamankan akses ke kotak penyimpanan.

2. Peningkatan Akurasi dan Kecepatan: Untuk meningkatkan performa *MoneyBox Plus*, dapat dilakukan penelitian lebih lanjut dalam mengoptimalkan algoritma pengolahan citra digital dan pemrosesan sensor warna. Hal ini dapat menghasilkan tingkat akurasi yang lebih tinggi dan waktu respons yang lebih cepat dalam mendeteksi dan mengenali pecahan uang.
3. Pengembangan Aplikasi *Mobile*: Mengembangkan aplikasi mobile yang terhubung dengan *MoneyBox Plus* dapat memberikan kemudahan bagi pengguna dalam memantau jumlah uang yang disimpan, melacak riwayat transaksi, atau mengatur pengaturan khusus. Aplikasi ini dapat memanfaatkan konektivitas Bluetooth atau Wi-Fi untuk berkomunikasi dengan *MoneyBox Plus*.
4. Skalabilitas: *MoneyBox Plus* dapat dirancang dengan kemampuan skalabilitas, sehingga pengguna dapat menambahkan modul atau komponen tambahan sesuai dengan kebutuhan. Misalnya, menambahkan modul sensor tambahan untuk mendeteksi dan memisahkan koin, atau memperluas kapasitas penyimpanan uang dengan modul tambahan.
5. Pengembangan Model Bisnis: *MoneyBox Plus* dapat dijadikan sebagai produk yang dapat dipasarkan secara komersial. Dalam bab ini, dapat diberikan saran tentang strategi pemasaran, analisis pasar, dan potensi kerjasama dengan lembaga keuangan atau toko retail untuk memperluas distribusi dan meningkatkan adopsi produk.

Saran-saran tersebut diharapkan dapat membantu dalam mengembangkan *MoneyBox Plus* menjadi produk yang lebih baik, lebih inovatif, dan dapat memberikan manfaat yang lebih besar bagi pengguna.

DAFTAR PUSTAKA

- Aguskhumaidi. (2019, september 5). *Mikrokontroler Arduino*. Retrieved from Agus Khumaidi, S.ST.,M.T.: <https://lecturer.ppns.ac.id/aguskhumaidi/2019/09/05/mikrokontroler-arduino/>

- Aji, H. (2018, Desember 16). *Pengertian, Perbedaan Power Supply AT dan ATX*. Retrieved from IlmuOnline.net:
<https://ilmuonline.net/pengertian-perbedaan-power-supply-at-dan-atx/>
- Anantajaya, I. M. (2021). REVIEW APLIKASI SENSOR PADA SISTEM . *Jurnal SPEKTRUM*, vol 8.
- ARIANSYAH, R. (2019). RANCANG BANGUN ALAT SORTIR JERUK NIPIS.
- Budiarso, Z. (2022). Penyortir Bola Berwarna Berbasis Arduino Menggunakan Metode Fuzzy. *Jurnal Teknik Komputer AMIK BSI*.
- Chairul Gunawan, F. N. (2021). Prototipe Light Meter Fotografi Studio Menggunakan Mikrokontroler . *JURNAL MEDIA INFORMATIKA BUDIDARMA*, 769-778.
- Dewi Lestari, M. H. (2019). RANCANG BANGUN SISTEM SORTIR BERDASARKAN WARNA . *jurutera*, 15.
- DIMAS K, R. D. (2018). PENYORTIR UANG KOIN BERBASIS ARDUINO UNO. *Universitas Gajah Mada*.
- ditempel. (2021, maret 19). *Menggunakan Modul LCD 16 x 2*. Retrieved from ditempel.com:
<https://www.ditempel.com/2021/03/menggunakan-modul-lcd-16-x-2.html>
- Februariyant, H. (2022). Penyortir Bola Berwarna Berbasis Arduino Menggunakan Metode Fuzzy. *Jurnal Teknik Komputer AMIK BSI*.
- Gramedia, B. (2021, Agustuts 21). *Pengertian Uang: Fungsi, Ragam, dan Teori Nilai Uang*. Retrieved from Pengertian Apa Itu Uang:
https://www.gramedia.com/literasi/uang/#Pengertian_Uang_-_Apa_itu_Uang
- Gudang, M. (2022, 06 19). *Pengertian dan Cara Sortir Barang dengan Mudah di Gudang*. Retrieved from Sortir Barang adalah:
<https://crewdbible.com/edukasi/topik/manajemen-gudang/sortir-barang-adalah#:~:text=Sortir%20barang%20adalah%20kegiatan%20pemisahan,seperti%20pemisahan%20sesuai%20alamat%20tujuan>
- Hanafi, M. H. (2019). Rancang Bangun Sistem Sortir Berdasarkan Warna Permen. *jurutera*.
- Ikhsan, M. A., Yahya, M., & Fiolana, F. A. (2018). Pendeteksi Kekeuhan Air Di Tandon Rumah Berbasis Arduino Uno. *Jurnal Qua Teknika*, Vol. 8 No. 2, 17-29.
- Irvawansyah, I. (2020). Rancang Bangun Media Pembelajaran Penyortiran Benda Berbasis Mikrokontroler. *Joule (Jurnal of Elictrical Engineering)*.
- Jufriyanto, J. (2020). Rancang Bangun Media Pembelajaran Penyortiran Benda Berbasis Mikrokontroler. *Joule (Journal of Electrical Engineering)*.
- Lestari, D. (2019). Rancang Bangun Sistem Sortir Berdasarkan Warna Permen. *jurutera*.
- Listiyono, H. (2022). Penyortir Bola Berwarna Berbasis Arduino Menggunakan Metode Fuzzy. *Jurnal Teknik Komputer AMIK BSI*.
- Mustafa, S. (2020). Rancang Bangun Media Pembelajaran Penyortiran Benda Berbasis Mikrokontroler. *joule (Jurnal of Elictrical Engineering)*.
- Nanda, T. R. (2018). Perancangan Sistem Sortir Buah Kopi Berdasarkan Warna Dengan Teknik Citra Digital Berbasis Mikrokontroler Atmega 328p. *KITEKTRO*.
- Ningsih. (2022). *Apa Yang Dimaksud Dengan Sensor? : Pengertian, Fungsi, Jenis-Jenis, Faktor dan Contoh Penerapan*. IDMETSFORA.
- Purnomo. (2017). Model Prototyping Pada Pengembangan Sistem Informasi. *Jurnal Informatika Merdeka pasuruan*, 55.
- Rijali, M., & Khana, J. R. (2020). Bangun Sistem Pemantauan Dan Pengendalian Cairan Infus Melalui Display Kontrol Dan Aplikasi Mobile Di Masa Pandemic Covid-19. *Ejournal Kajian Teknik Elektro Vol.5 No.1*, 1-21.
- Saputro, R. A. (2022). Penyortir Bola Berwarna Berbasis Arduino Menggunakan Metode Fuzzy. *Jurnal Teknik Komputer AMIK BSI*.
- Setiawan, R. (2021, Agustus 4). *Flowchart Adalah: Fungsi, Jenis, Simbol, dan Contohnya*. Retrieved from Dicoding:
<https://www.dicoding.com/blog/flowchart-adalah/>
- Simatupang, J. W., Prasetyo, B., Galina, M., & Suhartomo, A. (2022). Prototipe Mesin Penjual Air Mineral Otomatis berbasis Arduino Mega 2560 dan RFID-RC522. *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika Vol. 10 No. 2*, 484-499.
- Syaryadh, M. (2018). Perancangan Sistem Sortir Buah Kopi Berdasarkan Warna Dengan Teknik Citra Digital Berbasis Mikrokontroler Atmega 328p. *KITEKTRO*.

- T. W. Wisjhnuadji, A. N. (2020). Sistem Sortir Barang Otomatis Berbasis Arduino dengan Sensor . *Faktor Exacta*, 106.
- Yohanes C Saghoa, S. R. (2018). Kotak Penyimpanan Uang Berbasis . *Jurnal Teknik Elektro dan Komputer* , 167.
- Zulhelmi, Z. (2018). Perancangan Sistem Sortir Buah Kopi Berdasarkan Warna Dengan Teknik Citra Digital Berbasis Mikrokontroler Atmega 328p. *KITEKTRO*.
- Zulkarnain, M. (2020). Rancang Bangun Media Pembelajaran Penyortiran Benda Berbasis Mikrokontroler. *joule (Jurnal of Elictrical Engineering)*.