BAB IV
HASIL DAN PEMBAHASAN

4.1 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 :

4.1.1 Pengujan Menu Utama

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

Gambar 4. 1 Menu Utama *MoneyBox Plus*

4.1.2 Penerapan Cara Kerja Deteksi Warna

Berikut adalah langkah-langkah penerapan cara kerja deteksi warna pada Kotak Penyimpanan Uang (MoneyBox Plus) secara sistematis:

1. Langkah pertama adalah menghubungkan alat *MoneyBox Plus* ke *Power Supply*. Alat *Power Supply* ini menggunakan kabel data USB untuk memberikan daya.
2. Setelah *MoneyBox Plus* terhubung dengan *Power Supply*, mikrokontroler Arduino sebagai pusat pengendali akan mulai beroperasi untuk menjalankan mesin pada *MoneyBox Plus*.
3. Kemudian, menu utama *MoneyBox Plus* akan aktif dan ditampilkan pada layar.
4. Tekan tombol yang terletak di layar pada bagian "*Scan* Uang" untuk memulai proses pemindaian uang.
5. Selanjutnya, masukkan uang pecahan Rp. 50.000 atau Rp. 100.000 yang ingin disimpan ke dalam MoneyBox Plus. Dalam satu fase, batas maksimum memasukkan uang adalah 5 kali. Setelah itu, *MoneyBox Plus* akan menampilkan total uang yang telah di*scan* melalui proses deteksi warna.

Dengan mengikuti langkah-langkah ini, pengguna dapat secara sistematis mengoperasikan MoneyBox Plus dengan menggunakan fitur deteksi warna untuk menyimpan uang secara efektif dan terkontrol.

4.1.3 Pengujian Sensor Warna

Pada tahap pengujian sensor warna TCS34725, dilakukan pengambilan data warna yang telah diuji. Hasil

pengujian ini direpresentasikan dalam bentuk tabel yang berisi data warna yang terdeteksi oleh sensor. Tabel tersebut mencakup informasi mengenai berbagai parameter warna, seperti tingkat kecerahan, komponen merah, hijau, dan biru (RGB), serta data lainnya yang relevan. Data tersebut dapat digunakan untuk analisis lebih lanjut, seperti pengenalan pola warna atau penyesuaian parameter sensor.

Dengan adanya tabel hasil pengujian, peneliti dapat melihat dan menganalisis data warna yang terdeteksi oleh sensor TCS34725 secara sistematis. Hal ini membantu dalam memvalidasi kinerja sensor warna dan memastikan bahwa sensor dapat memberikan hasil yang akurat dan konsisten dalam mendeteksi warna pada MoneyBox Plus.

Tabel 4. 1 Pengujian Sensor Warna TCS34725

Data Uang	Data Warna Yang Terdeteksi					
	R		G		B	
	Batas Awal	Batas Akhir	Batas Awal	Batas Akhir	Batas Awal	Batas Akhir
Rp.50.000	48	70	102	117	86	110
Rp.100.000	75	87,5	83	95	67,5	88,5

Berdasarkan Tabel 4.1 diatas dapat terlihat bahwa pengujian yang dilakukan menghasilkan perubahan nilai RGB yang berbeda pada setiap object yang terdeteksi pada alat ini yaitu pada mata uang Rp.50.000 dan Rp.100.000 .

4.1.4 Uji Ukur Presentase Warna RGB

Pengujian presentase warna RGB pada alat ini dilakukan dengan metode langsung, yang dimana nantinya nilai alat yang ada pada tabel 4.1 akan dihitung secara keseluruhan

pada nilai keseluruhan warna yang telah disesuaikan yaitu 255.

Tabel 4. 2 Uji Ukur Presentase Warna RGB (%)

Data Uang	Data Warna Yang Terdeteksi					
	R		G		B	
	Batas Awal	Batas Akhir	Batas Awal	Batas Akhir	Batas Awal	Batas Akhir
Rp.50.000	18,8	27,4	40	45,8	33,7	43,13
Rp.100.000	29,4	34,3	32,5	37,2	26,4	34,70

Dari table 4.2 diatas, akan menghasilkan presentase warna RGB yang didapat setelah melakukan perhitungan antara hasil uji dengan warna standart yang telah ditentukan dengan rumus sebagai berikut :

(Jumlah Bagian : Jumlah Keseluruhan) *x* 100% = Presentase (%)

1. Uang Rp.50.000,00-

a. Batas Awal

%R = (48:255) x 100% = 18,8%
%G = (102:255) x 100% = 40%
%B = (86:255) x 100% = 33,7%

b. Batas Akhir

%R = (87,5:255) x 100% = 27,4%
%G = (95:255) x 100% = 45,8%
%B = (110:255) x 100% = 43,13%

2. Uang Rp.100.000,00-

a. Batas Awal

%R = (75:255) x 100% = 29,4%

%G = (83:255) x 100% = 32,5%

%B = (67,5:255) x 100% = 26,4%

b. Batas Akhir

%R = (87,5:255) x 100% = 34,3%

%G = (95:255) x 100% = 37,2%

%B = (88,5:255) x 100% = 34,70%

4.2 Hasil Penelitian

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

4.2.1 Layout Kode Menu Utama

```
void Menu_display()
{
  tft.setTextSize(3);
  tft.setTextColor(BLACK);
  tft.setCursor(40, 37);
  tft.print("MoneyBox");

  tft.setTextSize(3);
  tft.setTextColor(BLACK);
  tft.setCursor(185, 22);
  tft.print("+");

  DrawButtonScanUang();
  DrawButtonCekJumlah();
  DrawButtonAksesMasuk();
  DrawButtonResetPIN();
  DrawButtonLupaPIN();
  DrawButtonClear();
}
```

Gambar 4. 2 Layout Kode Menu Utama

Pada Gambar 4.2 Kode di atas adalah sebuah fungsi dalam program Arduino yang disebut "*Menu_display*". Fungsi ini bertanggung jawab untuk menampilkan menu utama pada layar menggunakan TFT (Thin-Film Transistor) display. Berikut adalah penjelasan mengenai setiap baris kode dalam fungsi "Menu_display":

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 `*DrawButtonScan*Uang()`, `*DrawButton*CekJumlah()`, `*DrawButton*AksesMasuk()`, `*DrawButtonReset*PIN()`, `*DrawButton*LupaPIN()`, dan `*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.

4.2.2 Layout Kode Tombol *Scan* Uang

```
void DrawButtonScanUang()
{
  tft.fillRoundRect(18, 95, 95, 70, 10, BLACKM);
  tft.fillRoundRect(20, 97, 91, 66, 10, JADE);
  tft.setTextSize(2);
  tft.setTextColor(WHITE);
  tft.setCursor(42, 110);
  tft.print("SCAN");
  tft.setCursor(42, 135);
  tft.print("UANG");
}

void DrawButtonScanUangPress()
{
  tft.fillRoundRect(18, 95, 95, 70, 10, BLACKM);
}
```

Gambar 4. 3 *Source Code Button Scan Uang*

Pada Gambar 4.3 memiliki penjelasan yang dimana Fungsi "DrawButton*Scan*Uang" dan "DrawButton*Scan*UangPress" adalah fungsi-fungsi dalam program Arduino yang bertanggung jawab untuk menggambar tombol "*Scan* Uang" pada layar menggunakan TFT display. Berikut adalah penjelasan mengenai setiap baris kode dalam fungsi "DrawButton*Scan*Uang":

1. `tft.*fillRound*Rect(18, 95, 95, 70, 10, BLACKM);`: Menggambar kotak dengan sudut melengkung (round rectangle) pada posisi (18, 95) dengan lebar 95 dan tinggi 70. Kotak ini akan diisi dengan warna hitam (BLACKM).

2. `tft.*fillRound*Rect(20, 97, 91, 66, 10, JADE);`: Menggambar kotak dengan sudut melengkung (round rectangle) yang sedikit lebih kecil daripada kotak sebelumnya. Kotak ini berada di dalam kotak sebelumnya dan akan diisi dengan warna JADE.
3. `tft.*setTextSize*(2);`: Mengatur ukuran teks menjadi 2 kali ukuran default.
4. `tft.*setTextColor*(*WHITE*);`: Mengatur warna teks menjadi putih.
5. `tft.*setCursor*(42, 110);`: Mengatur posisi kursor teks pada layar dengan koordinat (42, 110).
6. `tft.print("*SCAN*");`: Mencetak teks "*SCAN*" pada layar pada posisi yang telah ditentukan sebelumnya.
7. `tft.*setCursor*(42, 135);`: Mengatur posisi kursor teks pada layar dengan koordinat (42, 135).
8. `tft.print("UANG");`: Mencetak teks "UANG" pada layar pada posisi yang telah ditentukan sebelumnya.

Fungsi "DrawButton*Scan*UangPress" memiliki perbedaan dengan fungsi "DrawButton*Scan*Uang" pada baris kode berikut:

1. `tft.fillRoundRect(18, 95, 95, 70, 10, BLACKM);`: Menggambar kotak dengan sudut melengkung yang sama seperti fungsi "*DrawButtonScan*Uang", namun tidak diisi dengan warna apapun.

Fungsi "*DrawButtonScan*UangPress" digunakan untuk menggambar tampilan tombol "*Scan* Uang" saat tombol tersebut ditekan atau diberi efek tekanan. Dalam hal ini, hanya kotak tombol yang digambar tanpa ada teks yang ditampilkan.

4.2.2 Layout Kode Data Variable Sensor Warna

```
void SensorWarna()
{
  uint16_t clear, red, green, blue;
  tcs.getRawData(&red, &green, &blue, &clear);
  // Konversi data warna ke nilai RGB
  float r, g, b;
  tcs.getRGB(&r, &g, &b);
  float r1, g1, b1;
  tcs.getRGB(&r1, &g1, &b1);
  r1 = r1*2;
  g1 = g1*2;
  b1 = b1*2;
  nol = 0;
  limapuluh = 0;
  seratus = 0;
  tft.setTextSize(2);
  tft.setTextColor(BLACK);
  tft.setCursor(71, 32);
  tft.println("Silahkan");
  tft.setCursor(40, 53);
  tft.println("Masukkan Uang");
```

Gambar 4. 4 *Variable Source Code* Sensor Warna

Penjelasan pada Gambar 4.4 adalah kode dari program Arduino yang berfungsi untuk mengambil data warna dari sensor TCS34725. Berikut adalah penjelasan mengenai setiap baris kode tersebut:

1. `void SensorWarna()`: Ini adalah deklarasi fungsi dengan nama SensorWarna, yang berarti fungsi ini akan digunakan untuk melakukan operasi terkait sensor warna.

2. `uint16_t clear, red, green, blue;`: Ini adalah deklarasi variabel dengan tipe data `uint16_t` yang akan digunakan untuk menyimpan data warna yang diperoleh dari sensor. Variabel `clear` digunakan untuk menyimpan data intensitas cahaya yang terdeteksi, sedangkan variabel `red`,

`green`, dan `blue` digunakan untuk menyimpan data intensitas warna merah, hijau, dan biru.

3. `tcs.getRawData(&red, &green, &blue, &clear);`: Ini adalah perintah untuk mengambil data warna dari sensor menggunakan metode `getRawData()`. Data yang diperoleh akan disimpan di variabel `red`, `green`, `blue`, dan `clear`.

4. `float r, g, b;`: Ini adalah deklarasi variabel dengan tipe data `float` yang akan digunakan untuk menyimpan data warna dalam bentuk nilai RGB (merah, hijau, biru).

5. `tcs.getRGB(&r, &g, &b);`: Ini adalah perintah untuk mengambil data warna RGB yang sudah dikonversi dari sensor menggunakan metode `getRGB()`. Data RGB yang diperoleh akan disimpan di variabel `r`, `g`, dan `b`.

6. `float r1, g1, b1;`: Ini adalah deklarasi variabel dengan tipe data `float` yang akan digunakan untuk mengubah nilai RGB dengan faktor 2.

7. `tcs.getRGB(&r1, &g1, &b1);`: Ini adalah perintah untuk mengambil data warna RGB yang sudah dikonversi dengan faktor 2 dari sensor menggunakan metode `getRGB()`. Data RGB yang diperoleh dengan faktor 2 akan disimpan di variabel `r1`, `g1`, dan `b1`.

8. `r1 = r1*2; g1 = g1*2; b1 = b1*2;`: Ini adalah perintah untuk mengalikan nilai RGB dengan faktor 2 sehingga mendapatkan nilai yang ditingkatkan.

9. `nol = 0; limapuluh = 0; seratus = 0;`: Ini adalah deklarasi dan inisialisasi variabel dengan nilai 0 yang akan digunakan dalam penghitungan jumlah uang berdasarkan warna.

10. `tft.setTextSize(2);`: Ini adalah perintah untuk mengatur ukuran teks pada layar.

11. `tft.setTextColor(BLACK);`: Ini adalah perintah untuk mengatur warna teks menjadi hitam.

12. `tft.setCursor(71, 32);`: Ini adalah perintah untuk mengatur posisi kursor pada layar.

13. `tft.println("Silahkan");`: Ini adalah perintah untuk menampilkan teks "Silahkan" pada layar.

14. `tft.setCursor(40, 53);`: Ini adalah perintah untuk mengatur posisi kursor pada layar.

15. `tft.println("Masukkan Uang");`: Ini adalah perintah untuk menampilkan teks "Masukkan Uang" pada layar.

Kode di atas menggambarkan bagaimana sensor warna TCS34725 digunakan untuk mendapatkan data warna dan kemudian menampilkannya pada layar menggunakan perangkat TFT. Selain itu, kode juga melakukan konversi dan pengolahan data warna untuk keperluan selanjutnya, seperti penghitungan jumlah uang berdasarkan warna yang akan dimasukkan ke dalam MoneyBox Plus.

4.2.2 Layour Kode Penerapan Sensor Warna

```
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 4. 5 *Source code* warna pada uang 50.000

Kode yang ada pada gambar 4.5 merupakan sebuah kondisi (if statement) yang mengevaluasi nilai-nilai warna RGB (r, g, b) untuk mendeteksi warna tertentu. Jika nilai-nilai warna tersebut memenuhi kriteria yang ditentukan dalam kondisi, maka blok kode di dalamnya akan dieksekusi. Dalam kondisi ini, terdapat beberapa perbandingan yang dilakukan untuk setiap komponen warna (merah, hijau, dan biru). Jika nilai r berada di antara 48 dan 70, nilai g berada di antara 102 dan 117, dan nilai b berada di antara 86 dan 110, maka kondisi dianggap terpenuhi.

Jika kondisi tersebut terpenuhi, maka blok kode di dalam kurung kurawal akan dieksekusi. Pada blok kode tersebut, beberapa perintah digunakan untuk mengatur tampilan pada layar TFT, mengeset nilai variabel `limapuluh` menjadi 50, menggerakkan servo pada pin 28 ke posisi 65, menunggu selama 3 detik dengan perintah `delay(3000)`, dan mengembalikan posisi servo ke `position1`.

Secara keseluruhan, blok kode ini digunakan untuk mendeteksi dan menangani masukan uang pecahan Rp.50.000. Jika warna yang terdeteksi pada sensor sesuai dengan kriteria yang diberikan, maka tampilan pada layar TFT akan disesuaikan dengan informasi uang pecahan yang masuk, variabel `limapuluh` akan diubah menjadi 50, servo akan digerakkan, dan setelah beberapa detik servo akan kembali ke posisi awal.

Perlu diperhatikan bahwa potongan kode ini mungkin hanya merupakan contoh dari bagian yang lebih besar dari program MoneyBox Plus, yang mencakup pengenalan warna untuk beberapa pecahan uang yang berbeda dan reaksi yang sesuai terhadap deteksi tersebut.

```
else if (r1>150 && r1<175 && g1>166 && g1<190 && b1>135 && b1<177)
{
  tft.fillRoundRect(15, 100, 210, 80, 5, BLACKM);
  tft.fillRoundRect(17, 102, 206, 76, 5, SERATUS);

  tft.setCursor(31, 129);
  tft.setTextSize(3);
  tft.setTextColor(WHITE);
  tft.println("Rp.100.000");
  seratus = 100;
  servo_30.write(65);
  delay(3000);
  servo_30.write(position2);
}
```

Gambar 4. 6 *Source code* warna pada uang 100.000

Potongan kode pada gambar 4.6 merupakan bagian dari blok kondisional (if-else) yang berfungsi untuk mendeteksi warna tertentu pada sensor dan melakukan tindakan yang sesuai. Kondisi ini adalah cabang kedua dari blok kondisional, yang akan dievaluasi jika kondisi sebelumnya tidak terpenuhi.

Pada kondisi ini, nilai-nilai warna RGB yang baru (r1, g1, b1) dievaluasi. Jika nilai r1 berada di antara 150 dan 175, nilai g1 berada di antara 166 dan 190, dan nilai b1 berada di antara 135 dan 177, maka kondisi dianggap terpenuhi. Jika kondisi tersebut terpenuhi, blok kode di dalamnya akan dieksekusi. Kode tersebut mirip dengan kode pada kondisi sebelumnya. Tampilan pada layar TFT akan disesuaikan dengan informasi uang pecahan yang masuk (Rp.100.000), variabel `seratus` akan diubah menjadi 100, servo pada pin 30 akan digerakkan ke posisi 65, dilakukan penundaan selama 3 detik dengan perintah `delay(3000)`, dan akhirnya servo akan kembali ke posisi awal (`position2`).

Dengan adanya kondisi ini, sistem dapat mendeteksi dan menangani masukan uang pecahan Rp.100.000. Jika warna yang terdeteksi pada sensor sesuai dengan kriteria yang diberikan, maka tampilan pada layar TFT akan disesuaikan dengan informasi uang pecahan yang masuk, variabel `seratus` akan diubah menjadi 100, servo akan digerakkan, dan setelah beberapa detik servo akan kembali ke posisi awal. Perlu diingat bahwa potongan kode ini mungkin hanya merupakan contoh dari bagian yang lebih besar dari program MoneyBox Plus, yang mencakup pengenalan warna untuk beberapa pecahan uang yang berbeda dan reaksi yang sesuai terhadap deteksi tersebut.

```
else
{
  tft.fillRoundRect(15, 100, 210, 80, 5, BLACKM);
  tft.fillRoundRect(17, 102, 206, 76, 5, WHITE);

  tft.setCursor(85, 129);
  tft.setTextSize(3);
  tft.setTextColor(BLACK);
  tft.println("Rp.0");
  nol = 0;
  servo.write(0);
}
```

Gambar 4. 7 Source code tidak ada uang masuk

Blok kode pada gambar 4.7 merupakan bagian dari blok kondisional `else`, yang akan dieksekusi jika tidak ada kondisi sebelumnya yang terpenuhi. Artinya, jika warna yang terdeteksi pada sensor tidak sesuai dengan kriteria warna yang telah ditentukan sebelumnya untuk pecahan uang Rp.50.000 dan Rp.100.000, maka blok kode ini akan dieksekusi. Pada blok kode ini, tampilan pada layar TFT akan disesuaikan dengan informasi bahwa uang yang terdeteksi adalah Rp.0 atau tidak ada uang yang masuk. Kotak tampilan pada layar TFT akan diisi dengan warna latar belakang hitam dan warna frame putih. Informasi "Rp.0" akan ditampilkan di tengah kotak dengan menggunakan ukuran teks yang lebih besar. Variabel `nol` akan diubah menjadi 0 untuk menandakan bahwa tidak ada uang yang terdeteksi. Selain itu, servo pada pin yang sesuai akan digerakkan ke posisi awal (0) untuk menunjukkan bahwa tidak ada aksi yang perlu dilakukan terkait dengan uang yang masuk.

Blok kode ini memberikan respons yang sesuai jika warna yang terdeteksi pada sensor tidak cocok dengan warna yang diharapkan untuk pecahan uang tertentu. Ini bisa terjadi jika warna yang terdeteksi tidak termasuk dalam kriteria yang telah ditentukan sebelumnya. Dengan adanya blok kondisional `else`, sistem dapat mengenali situasi ketika tidak ada pecahan uang yang terdeteksi dan meresponsnya secara tepat dengan menampilkan informasi "Rp.0" dan menggerakkan servo ke posisi awal.

4.3 Hasil Uji Coba

Setelah melakukan uji coba terhadap sistem deteksi warna pada Kotak Penyimpanan Uang (MoneyBox Plus) berbasis mikrokontroler Arduino, didapatkan hasil yang diharapkan. Sensor warna TCS34725 mampu mengenali dengan akurat warna-warna yang telah ditentukan 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.Pada setiap pengujian, sensor warna bekerja dengan baik dalam mengenali warna dan memberikan respons sesuai dengan kondisi yang telah ditentukan. Hasil uji coba ini menunjukkan bahwa sistem deteksi warna pada MoneyBox Plus mampu mendukung fungsi pengenalan pecahan uang secara efektif.

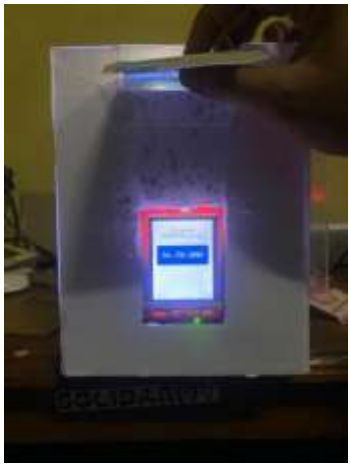
Selain itu, hasil uji coba juga memvalidasi kinerja layar TFT yang digunakan untuk menampilkan informasi mengenai pecahan uang yang terdeteksi. Informasi yang ditampilkan dengan jelas dan akurat, memudahkan

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

.4.3.1 Uji Coba 1

Uji coba tujuan nomor 1 : menampilkan hasil yang telah diajarkan kepada sensor untuk pecahan uang Rp.50.000,00 yang memiliki range warna biru bernama cerulean blue yang memiliki kode RGB dalam range R> 48 sampai R<70, G>102 sampai G<117, dan B>86 sampai B<110 yang dimana nantinya ketika uang melewati sensor, sensor akan mendeteksi kalua warna yang telah di inputkan tersebut telah terdefinisikan sebagai mata uang Rp.50.000,00



Gambar 4. 8 Uji Coba Uang Rp.50.000,00

Dengan di inputkan nya delay dengan waktu 3 detik, maka hal tersebut bisa digunakan untuk memproses ke tahap berikut nya atau memberikan jeda jarak untuk memasukkan uang berikutnya, dengan begitu proses *scan* uang tidak kebingungan dengan hasilnya.

4.3.2 Uji Coba 2

Uji coba tujuan nomor 2 : menampilkan hasil yang telah diajarkan kepada sensor untuk pecahan uang Rp.100.000,00 yang memiliki range warna biru bernama cerulean blue yang memiliki kode RGB dalam range R> 75 sampai R<87,5, G>83 sampai G<95, dan B>67,5 sampai B<88,5 dan karena disini angka yang dimasukkan sama dengan variable yang dimiliki pecahan uang Rp.50.000 maka disini semua angkar variable dari kode RGB dikalikan 2 yang menghasilkan R>150 sampai R<175, G>166 sampai G<190, dan B>135 sampai B<177 yang dimana nantinya ketika uang melewati sensor, sensor akan mendeteksi kalua warna yang telah di inputkan tersebut telah terdefinisikan sebagai mata uang Rp.100.000,00

Gambar 4. 9 Uji Coba Uang Rp.100.000,00

Dengan di inputkan nya delay dengan waktu 3 detik, maka hal tersebut bisa digunakan untuk memproses ke tahap berikut nya atau memberikan jeda jarak untuk memasukkan uang berikutnya, dengan begitu proses *scan* uang tidak kebingungan dengan hasilnya.