

## BAB IV HASIL DAN PEMBAHASAN

### 4.1 Implementasi Sistem

Implementasi berfungsi sebagai suatu langkah terakhir dalam penciptaan suatu objek. Hasil dari Implementasi ini adalah alat yang mudah di pahami dan di terapkan. Implementasi sortir buah alat ini tomat ini terdiri dari tiga bagian: perancangan kerangka sortir, perancangan konveyor Penyortir buah, dan rangkaian elektronika.

#### 4.1.1 Implementasi Kerangka Alat Sortir

Implementasi untuk kerangka alat pemisah yang mempergunakan bahan besi hollow aluminium dan juga *bahan plat stainless*, Aluminium di pergunakan digunakan di karenakan mudah di proses dan memiliki berat yang relative ringan, aluminium yang digunakan memiliki ukuran 1 x 2 cm, 2 x 4 cm dan *plat stainless* digunakan sebagai dasar alas konveyor alat sortir sehingga konveyor akan kokoh pada saat di lewati buah pada saat proses penyortiran

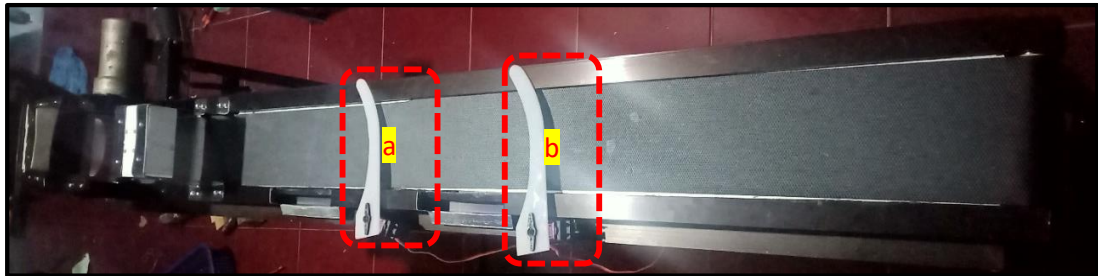
Rangka untuk alat pemisah berbentuk ruang bangun persegi panjang yakni dengan ukuran panjang 150 centimeter, untuk lebar konveyor 15 centimeter dan untuk tinggi konveyor 66 centimeter, Dalam proses mengerjakannya terdiri dari beberapa urutan, Langkah kesatu menggambar desain kerangka, kemudian hollow dipotong mempergunakan gerinda potong sesuai dengan ukuran desain. Langkah ke 2 menyambung hollow + plat stainless sebagai landasan belt konveyor sebagai dasar sehingga alas konveyor lebih kokoh pada saat di lalui buah pada saat sortir, Implementasi alat kerangka dapat dilihat pada gambar 4.1.



Gambar 4.1. Desain beserta Implementasi Kerangka untuk Alat pemisah/sortir

#### 4.1.2 Implementasi Akuator Pemisah/Penyortir Buah

Implementasi aktuator pemisah buah tomat memanfaatkan akrilik yang di potong sesuai dengan dasain,selanjutnya untuk bagian hopper dibikin menggunakan triplek berlapis stanles dan lubang berdiameter 9 cm yang berfungsi untuk mengeluarkan buah tomat pada saat awal/start proses penyortiran, Sesuai dengan peringatan yang telah di tentukan aktuator bertindak untuk memindahkan tomat kewadah yang sesuai .Aktuator penggerak menggunakan dua motor servo yang terhubung ke rangkaian arduino .Implementasi aktuator pemisah buah dapat dilihat pada gambar 4.2 (gambar a dan b). Gambar (a) berfungsi sebagai aktuator di depan, sedangkan gambar (b) berfungsi sebagai aktuator di belakang.



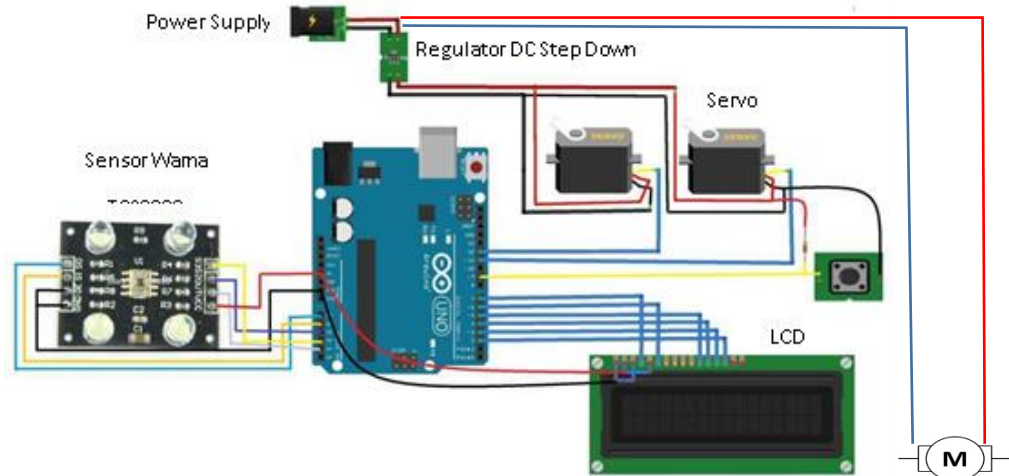
Gambar 4.2 .Implementasi Aktuator pemisah buah

#### 4.1.3 Implementasi Perangkat Lunak (*software*)

Program mikrokontroler Arduino dibuat menggunakan bahasa pemrograman C. Perangkat lunak editor dan kompiler digunakan untuk membuat program. File \*.ino yang telah dirakit oleh perangkat lunak file akan diunggah ke mikrokontroler Arduino, Penulisan Kode/Script yang ditanamkan (embed) Pada *Software* IDE Arduino Dapat dilihat Pada Gambar 4.3 dengan Kode/Script Pemogramran Microcontroller Arduino Secara Keseluhan.

## 4.2 Pengujian Alat

Pengujian/percobaan bertujuan untuk menemukan kekurangan pada alat dan memastikan apakah alat yang telah di buat berjalan dengan baik. Pengujian alat yang dilakukan meliputi pengujian sensor warna TCS3200, actuator penyeleksi buah tombol kendali dan pengujian alat secara keseluruhan.



Gambar 4.4 .Rangkaian Sistem Keseluruhan

## 4.3 Pengujian Sensor Warna TC3200

Untuk menentukan apakah rangkaian unit sensor TCS3200 dapat beroperasi dengan aman dan dapat menampilkan data RGB pada LCD dan *monitor serial*, percobaan dilakukan. Percobaan dilakukan dengan menghubungkan pin keluaran peringatan sensor ke pin Arduino analog (A0, A1, A2, A3, A4). Program yang tercantum di bawah ini adalah untuk kalibrasi sensor peringatan TCS3200.

```
#include <Wire.h> // i2C Connection Library
#include <LiquidCrystal_I2C.h> //i2C LCD Library
#include <Servo.h> //i2C LCD Library
LiquidCrystal_I2C lcd(0x27, 16, 2);
//LiquidCrystal_I2C lcd(0x3F, 16, 2);
Servo myser1;
Servo myser2;
#define S0 2
#define S1 3
#define S2 4
#define S3 7
#define sensorOut 8
```

```

int frequencyr = 0;
int frequencyg = 0;
int frequencyb = 0;
void setup() {
  myser1.attach(6);
  myser1.write(170);
  myser2.attach(5);
  myser2.write(170);
  pinMode(S0, OUTPUT);
  pinMode(S1, OUTPUT);
  pinMode(S2, OUTPUT);
  pinMode(S3, OUTPUT);
  pinMode(sensorOut, INPUT);
  // Setting frequency-scaling to 20%
  digitalWrite(S0,HIGH);
  digitalWrite(S1,LOW);
  lcd.begin();
  lcd.clear();
  lcd.noCursor();}
void loop() {
  lcd.setCursor(0, 0);
  // Setting red filtered photodiodes to be read
  digitalWrite(S2,LOW);
  digitalWrite(S3,LOW);
  // Reading the output frequency
  frequencyr = pulseIn(sensorOut, LOW);
  // Printing the value on the serial monitor
  lcd.setCursor(0, 0);
  lcd.print("R=");
  lcd.print(frequencyr);//printing RED color frequency
  lcd.print(" ");
  delay(100);
  // Setting Green filtered photodiodes to be read
  digitalWrite(S2,HIGH);
  digitalWrite(S3,HIGH);
  // Reading the output frequency
  frequencyg = pulseIn(sensorOut, LOW);
  // Printing the value on the serial monitor
  lcd.print("G=");
  lcd.print(frequencyg);//printing RED color frequency

```

```

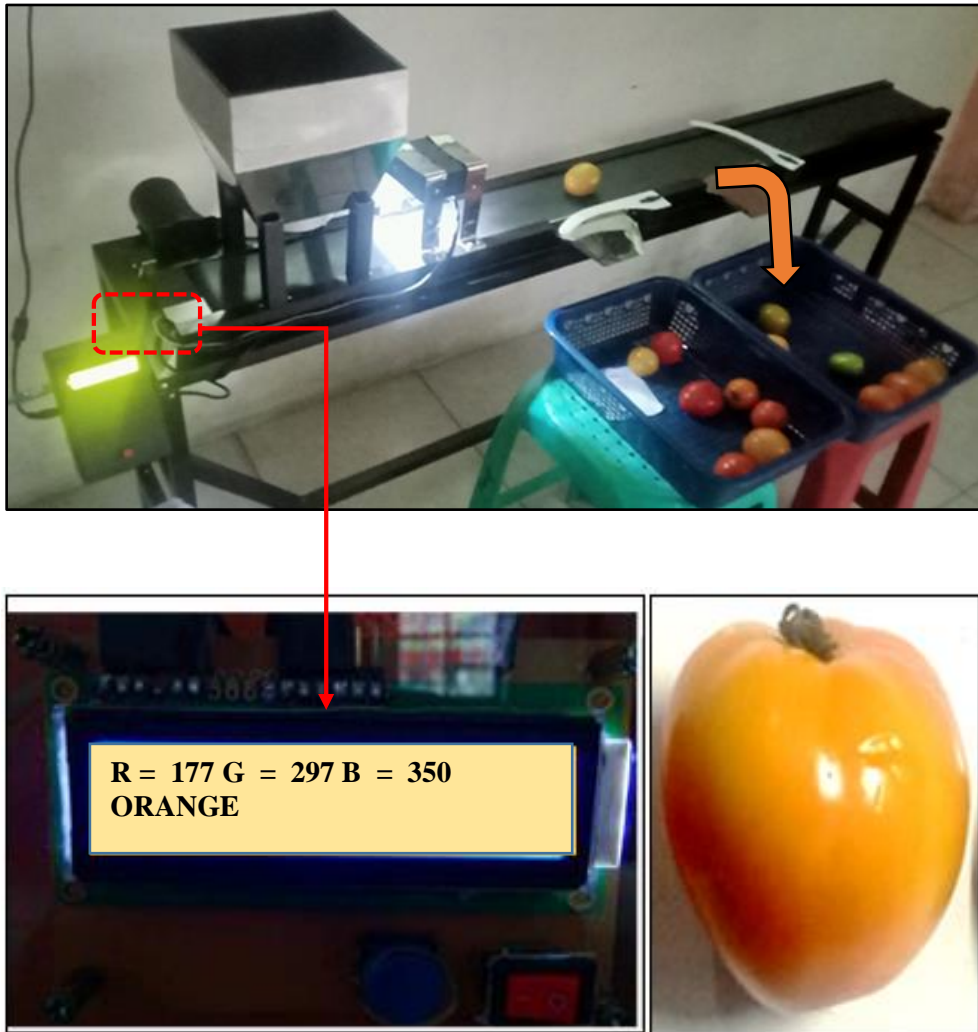
lcd.print(" ");
delay(100);
// Setting Blue filtered photodiodes to be read
digitalWrite(S2,LOW);
digitalWrite(S3,HIGH);
// Reading the output frequency
frequencyb = pulseIn(sensorOut, LOW);
// Printing the value on the serial monitor
lcd.print("B=");
lcd.print(frequencyb);//printing RED color frequency
lcd.println(" ");
delay(100);
if((frequencyr > 200)&&(frequencyr < 300)&&(frequencyg > 450)&&(frequencyg <
550)&&(frequencyb > 350)&&(frequencyb < 450)){
lcd.setCursor(0, 1);
lcd.println("MERAH ");
myser1.write(100);
delay(4000);
myser1.write(170);
delay(1000); }
else if((frequencyr > 100)&&(frequencyr < 250)&&(frequencyg > 250)&&(frequencyg <
350)&&(frequencyb > 300)&&(frequencyb < 400)) {
lcd.setCursor(0, 1);
lcd.println("ORANYE ");
myser2.write(100);
delay(7500);
myser2.write(170);
delay(1000);}
else if((frequencyr > 250)&&(frequencyr < 350)&&(frequencyg > 250)&&(frequencyg <
350)&&(frequencyb > 350)&&(frequencyb < 450)){
lcd.setCursor(0, 1);
lcd.println("HIJAU ");
delay(1000);}
else{
lcd.setCursor(0, 1);
lcd.println(" ");}

```

Diagram 4.1. Program Pengujian Sensor Warna

#### 4.4 Pengujian Sensor dengan Buah Tomat

Pada titik ini dalam prosedur, buah tomat dengan warna oranye, merah, dan hijau digunakan. Ketika tomat di posisikan dengan benar di atas konveyor, pembacaan data RGB (Merah, Hijau, dan Oranye) akan dilakukan dan akan terbaca di monitor LCD dan serial.



Gambar 4.5 .Percobaan Alat menggunakan buah tomat Oranye

Gambar 4.5. Hasil tampilan RGB Nilai menggunakan LCD dan serial monitor  
Dari gambar 4.5 kita dapat melihat bahwa kemampuan sensor TCS3200 untuk mendeteksi oranye sesuai dengan sampel buah tomat dengan daun berwarna hijau. Nilai RGB yang tidak selalu sesuai dengan kondisi ruangan

#### 4.4.1 Penguji Sensor Dengan buah Berwarna Merah

Selama sesi pengujian khusus ini, buah tomat dengan cangkang berwarna merah digunakan. Ketika tomat ditempatkan secara akurat di atas konveyor dan akan terbaca di layar, LCD dan monitor serial kemudian akan menampilkan data dalam warna RGB (Merah, Hijau, dan Oranye).



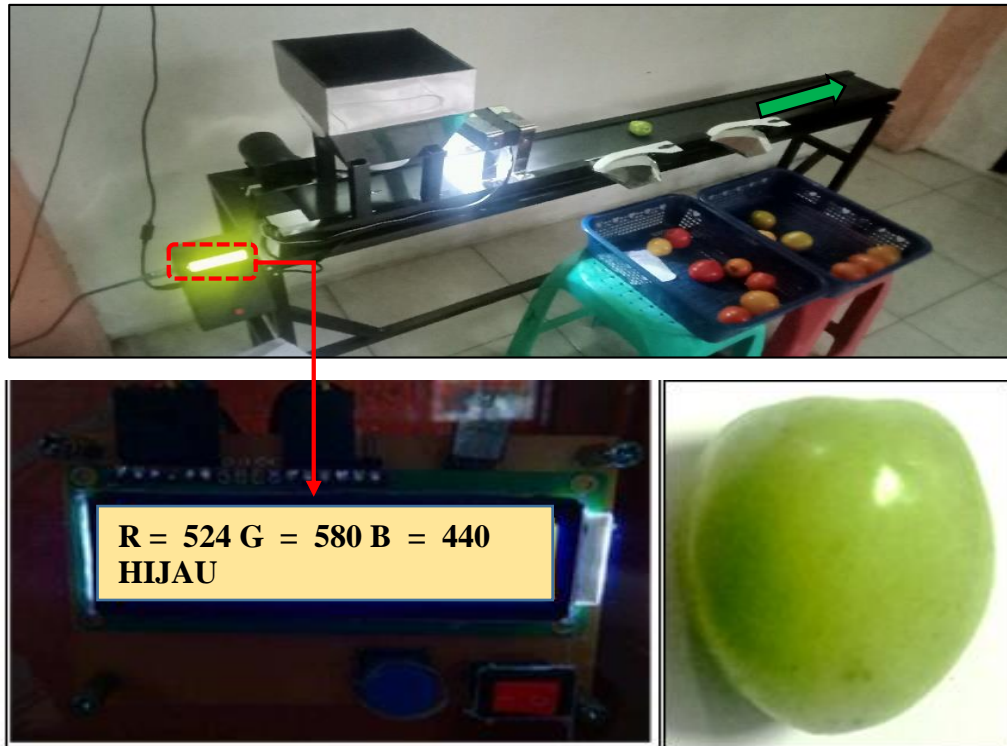
Gambar 4.6 .Percobaan alat dengan menggunakan buah Tomat berwarna Merah

Dari gambar 4.6 kita dapat melihat tampilan sensor TCS3200 yang dapat mendeteksi warna merah sesuai dengan sampel buah tomat yang memiliki pembacaan merah, dan bahwa Nilai RGB dalam gambar dapat berubah sesuai dengan kondisi cahaya diruangan.

#### 4.4.2 Pengujian Sensor dengan Buah tomat berwarna Hijau

Selama sesi pengujian khusus ini, buah tomat dengan tomat warna hijau. Ketika tomat ditempatkan secara akurat di atas sensor warna, data dalam ruang akan terbaca RGB (Merah, Hijau, dan Biru) kemudian akan ditampilkan pada LCD dan monitor serial.





Gambar 4.7.Percobaan alat dengan menggunakan buah Tomat berwarna Hijau

#### 4.4.3 Percobaan sensor dengan Buah

Selama sesi pengujian khusus ini, menggunakan buah tomat yang berwarna hijau, Ketika tomat diletakan dengan benar di atas sensor warna, data RGB (Merah, Hijau, dan Biru) akan ditampilkan pada LCD dan monitor serial.

#### 4.4.4 Pengujian Sistem Keseluruhan

Percobaan system secara keseluruhan adalah untuk mengumpulkan data yang akan digunakan untuk membuat sumber yang dapat digunakan sebagai ambang batas untuk mendeteksi berbagai jenis buah tomat selama proses ekstraksi buah.

Berikut beberapa data yang diambil dari hasil pembacaan warna buah tomat menggunakan sensor TCS3200.(Tomy,Fadi.2017)



Tabel 4.1. Hasil Uji Sensor TCS3200 dengan Tomat berwarna Oranye

No	Warna Tomat	Nilai Insensitas Pada LCD			Warna yang Terbaca	Waktu(ms)
		Red	Green	Blue		
1	Tomat warna Orange	177	297	350	Oranye	1062
2	Tomat warna Oranye	277	349	370	Oranye	1062
3	Tomat warna Oranye	220	344	370	Oranye	1062
4	Tomat warna Oranye	174	253	370	Oranye	1062
5	Tomat warna Oranye	176	328	340	Oranye	1062

Dari Tabel 4.1. dapat dilihat bahwa sensor akan mendeteksi objek yang berwarna oranye ketika intensitas Merah lebih tinggi daripada Hijau atau Biru atau ketika  $|R-G|$  lebih besar dari 50. Total waktu yang dibutuhkan untuk membaca dan mengkonsumsi satu buah tomat berwarna oranye adalah 1238 milidetik

Tabel 4.2. Hasil Pembacaan Sensor TCS3200 dengan Tomat Merah

No	Warna Tomat	Nilai Insensitas pada LCD			Warna Yang Terbaca	Waktu (ms)
		Red	Green	Blue		
1	Tomat warna Merah	265	503	410	Merah	1237
2	Tomat warna Merah	271	579	430	Merah	1237
3	Tomat warna Merah	229	503	430	Merah	1237
4	Tomat warna Merah	278	498	470	Merah	1237
5	Tomat warna Merah	259	498	400	Merah	1238

Dari data Tabel 4.2 dapat ditunjukkan bahwa sensor untuk mendeteksi warna Merah ketika nilai warna Merah memiliki intensitas warna di atas 100 dan lebih

intens daripada warna Hijau dan Biru. Jumlah total waktu yang dibutuhkan untuk membaca dan mengkalibrasi bendera peringatan untuk kehijauan adalah 1237 milidetik.

**Tabel 4.3. Hasil Sensor TCS3200 dengan Instalasi Tomat Berwarna Hijau**

No	Warna Tomat	Nilai Insensitas Pada LCD			Warna yang terbaca	Waktu (ms)
		Red	Green	Blue		
1	Tomat Warna Hijau	524	588	440	HIJAU	1237
2	Tomat Warna Hijau	524	596	450	HIJAU	1237
3	Tomat Warna Hijau	520	590	420	HIJAU	1237
4	Tomat Hijau	522	585	410	HIJAU	1236
5	Tomat Warna Hijau	524	590	450	HIJAU	1237

Dapat ditunjukkan dari Tabel 4.3

bahwa sensor untuk mendeteksi warna mendeteksi warna kontras tinggi ketika nilai Hijau memiliki intensitas warna yang lebih tinggi daripada nilai Merah dan Biru. Waktu yang dibutuhkan untuk membaca dan mengkalibrasi sinyal peringatan visibilitas tinggi adalah 1237 milidetik.

**Tabel 4.4 .Hasil Pengujian Sensor TCS3200 tanpa Buah Tomat**

No	Warna belt conveyor	Nilai Intensitas pada LCD			Warna yang Terdeteksi	Waktu(ms)
		Red	Green	Blue		
1	-	531	607	470	-	1063
2	-	531	607	470	-	1063
3	-	531	607	470	-	1063
4	-	531	607	470	-	1063
5	-	531	607	470	-	1063

Dari Tabel 4.4 di atas, dapat dilihat bahwa sensor untuk warna tidak dapat mendeteksi keberadaan tomat yang terletak di sebelah sensor TCS3200, sehingga

tingkat intensitas warna yang ada adalah nol. Waktu yang dibutuhkan untuk membaca dan mengkalibrasi adalah 1062 milidetik. (Abdul Haris. 2019)

Dari hasil percobaan menggunakan beberapa kali dari masing-masing metode, berwarna oranye, kuning kehijauan, hijau, dan tidak ada buah tomat yang digunakan dalam setiap sampel. Sensor cukup baik dalam mendeteksi sinyal alarm. Untuk meningkatkan keandalan hasil sistem peringatan, sensor akan mendeteksi peringatan buah tomat dengan tiga bagian pembacaan, kemudian menggunakan dua tingkat intensitas peringatan yang relatif rendah dan sering terjadi..

Waktu rata-rata yang dibutuhkan untuk menyortir satu buah tomat hingga buah tomat masuk ke dalam wadah yang ditentukan dan aktuator kembali ke posisi awal adalah 1238 ms. Ada tiga contoh RGB nilai pembaca dalam 1238 milidetik., Jadi sensor warna yang dapat membaca warna RGB dalam waktu  $\frac{128}{3} \text{ ms} = 412 \text{ ms}$

#### **4.5 Analisa Ketelitian Alat**

Untuk menentukan tingkat ketelitian pada ketinggian ini, pengujian pertama harus dilakukan sekitar 60 kali,

setelah itu kesalahan yang terjadi di ketinggian harus dinilai kembali. Tes ini dilakukan hingga lima kali untuk menentukan tingkat keparahan kesalahan pada perangkat. Hasil percobaan dapat dilihat pada lampiran. Anda dapat menggunakan rumus untuk mendeteksi kesalahan dalam setiap perhitungan berikutnyadari pengujian dihitung berapa banyak kesalahan yang terjadi pada alat, lalu dilakukan pengujian kedua sebanyak 60 kali, lalu dihitung kembali kesalahan yang terjadi pada alat. Demikian pengujian dilakukan hingga 5 kali, untuk mengetahui persentase error pada alat. Hasil pengujian dapat dilihat pada lampiran C. Untuk mendapatkan nilai error pada masing-masing percobaan dapat menggunakan rumus:

$$Error = \left| \frac{\text{Total Error}}{\text{Total Pengujian}} \right| \times 100\%$$

Berikut merupakan jumlah error dari hasil pengujian:

**Tabel 4.5 .Total Error Hasil Pengujian**

No	Jumlah Pengujian	Error	% Error
1	60 x	9	15
2	60 x	13	21,6
3	60 x	3	5
4	60 x	7	11,6
5	60x	8	13,3
	<b>Total</b>	<b>40</b>	<b>13,3%</b>

Dengan demikian dapat dijumlahkan akurasi alat pemisah dengan perhitungan berikut:

$$Akurasi = \left| \frac{Total\ Error - Total\ Penguji}{Total\ Pengujian} \right| \times 100\%$$

$$Akurasi = \left| \frac{(9+13+3+7+8) - 300}{300} \right| \times 100\%$$

$$Akurasi = \left| \frac{40-300}{300} \right| \times 100\%$$

$$Akurasi = \left| \frac{260}{300} \right| \times 100\%$$

$$Akurasi = 0,866 \times 100\%$$

$$Akurasi = 86,66 \%$$

Hasil perhitungan di yang di tampilkan di atas menunjukkan untuk akurasi dari alat pemisah tomat adalah sebesar 86,66%, dan tingkat error alat sortir sebesar 13,33%.

