

Fruit Color Sorter:
**Rancang Bangun Sistem Sortir Buah Tomat Berdasarkan
Warna RGB Menggunakan Arduino Uno R3**



IPB University
— Bogor Indonesia —

Disusun Oleh:
Kelompok 9

Zahra Aulia Firdausi	G64180030
Ulfainil Aisyah	G64180045
Putri Melanita Londong Bua	G64180053

**INSTITUT PERTANIAN BOGOR
BOGOR
2021**

I. Latar Belakang

Perkembangan teknologi membuat segala sesuatu menjadi lebih mudah, praktis, dan efisien. Salah satu teknologi yang biasa digunakan saat ini adalah teknologi mikrokontroler. Mikrokontroler merupakan alat atau sistem yang dapat membantu pekerjaan manusia. Sehingga, pekerjaan yang dulunya dilakukan secara manual bisa dilakukan secara otomatis.

Produksi hasil pertanian dan perkebunan juga berkembang pesat seiring berkembangnya zaman. Salah satu hasil pertanian indonesia, yaitu buah tomat. Salah satu tahap dalam proses pengolahan hasil pertanian dan perkebunan adalah penyortiran buah. Penyortiran bertujuan untuk memilah-milah buah tomat berdasarkan kualitas warna dan ukuran tertentu. Proses penyortiran buah tomat umumnya dilakukan secara manual menggunakan tenaga manusia. Penyortiran secara manual ini memiliki kelemahan, yaitu persepsi visual manusia yang berbeda, memerlukan tenaga manusia yang mengetahui tentang mutu buah.

Negara-negara maju telah menggunakan alat sortir otomatis, namun alat sortir ini masih tergolong mahal. Oleh karena itu, kami bertujuan untuk merancang sebuah sistem alat otomatis untuk menyortir buah tomat berdasarkan warna menggunakan sensor warna dan sensor berat untuk menghitung berat buah hasil sortir secara otomatis. Diharapkan penelitian ini dapat meningkatkan efisiensi dalam proses pengolahan hasil pertanian khususnya penyortiran kematangan buah.

II. Tujuan

Berdasarkan latar belakang yang sudah diuraikan tujuan dari penelitian ini, yaitu membuat sistem untuk memudahkan pemisahan buah yang sudah matang dan belum matang serta pembuatan alat untuk mengefisiensikan proses penyortiran.

III. Tinjauan Pustaka

A. Tomat

Tomat (*Lycopersicum esculentum*) adalah tanaman dari famili *Solanaceae* yang tumbuh secara alami di Amerika Tengah dan Selatan dari Meksiko hingga Peru. Kata “tomat” berasal dari bahasa Nawat, dimana tomat berkerabat dekat dengan kentang (Kusumaningtyas dan Asmara 2016).



Gambar 1 Buah Tomat

Berdasarkan warna dari buah tomat, kematangan dapat dikelompokan menjadi tiga bagian, yaitu:

1. Buah Tomat Matang

Buah tomat yang matang diidentifikasi dengan warna kulit oranye kemerahannya sampai dengan warna merah sempurna.



Gambar 2 Buah Tomat Matang

2. Buah Tomat Setengah Matang

Buah tomat yang setengah matang diidentifikasi dengan warna kulit hijau kekuningan dan oranye.



Gambar 3 Buah Tomat Setengah Matang

3. Buah Tomat Belum Matang

Buah tomat yang belum matang diidentifikasi dengan warna kulit hijau.



Gambar 4 Buah Tomat Belum Matang

B. Color Sensor TCS3200

Color sensor TCS3200 adalah konverter yang diprogram untuk mengubah warna menjadi frekuensi, yang tersusun atas konfigurasi fotodiode silikon dan konverter arus ke frekuensi dalam IC CMOS monolithic yang tunggal (Athifa dan Rachmat 2019). Keluaran dari sensor ini adalah gelombang kotak (*duty cycle 50%*) dengan frekuensi yang berbanding lurus dengan intensitas cahaya (*irradiance*) (*Programmable Color Light to Frequency Converter*).

C. Load Cell 5 Kg dan Modul HX711

Load cell adalah perangkat yang mengeluarkan sinyal listrik sebanding dengan gaya/beban yang diterima. Resistor yang bertanda T1 dan T2 merupakan Strain Gauge yang menerima gaya tarik (*tension*) saat *Load cell* menerima beban. Sedangkan Resistor yang ditandai dengan C1 dan C2 adalah pengukur regangan yang menerima gaya tekan (*compression*) ketika *Load cell* dibebani (Wicaksono dan Hidayat 2017).

D. Micro Servo Motor

Micro servo motor adalah sebuah motor dengan sistem umpan balik tertutup dimana posisi dari motor akan diinformasikan kembali ke rangkaian kontrol yang ada di dalam servo motor (Sopyan 2019).

E. Arduino Uno

Arduino Uno merupakan board mikrokontroler yang didasarkan pada ATmega328. Arduino Uno digunakan sebagai komunikasi serial dengan PC dan pengontrol sensor. Pada Arduino Uno, terdapat 14 pin digital *input/output* dan 6 diantaranya bisa digunakan sebagai PWM, 6 *Output* analog, *osilator* kristal 16 MHz, sebuah koneksi USB, *power jack*, ICSP header, dan tombol *reset* (Akbar 2016).

F. LCD

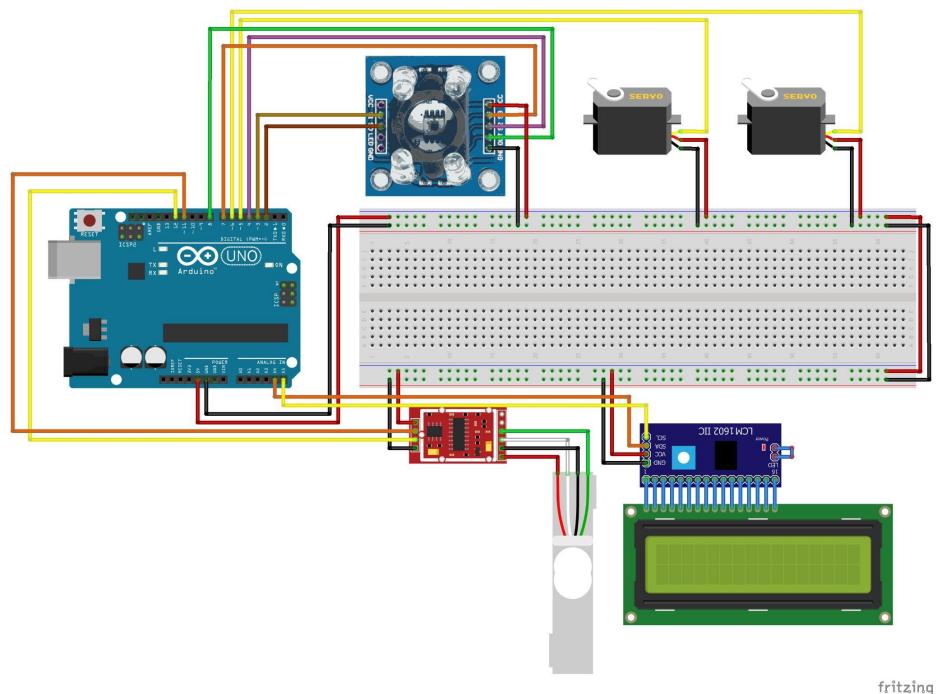
LCD (*Liquid Crystal Display*) merupakan salah satu display elektronika yang sering digunakan. LCD dibuat dengan CMOS *logic* yang bekerja dengan tidak menghasilkan cahaya melainkan memantulkan cahaya yang ada di sekitarnya terhadap *front-lit* atau mentransmisikan cahaya dari *back-lit* (Subagyo 2017). LCD dapat berfungsi untuk menampilkan suatu nilai hasil sensor, menampilkan teks, atau menampilkan menu pada aplikasi mikrokontroler (Simbar dan Syahrin 2017).

IV. Rancangan Sistem

A. Deskripsi Sistem

Fruit color sorter adalah sistem alat otomatis untuk menyortir buah tomat berdasarkan warna kematangannya menggunakan sensor warna. *Channel* warna yang digunakan adalah RGB (*Red, Green, Blue*). Selain alat sortir, terdapat alat juga untuk menghitung berat buah tomat yang telah disortir secara otomatis menggunakan sensor berat.

B. Desain Sistem



Gambar 5 Desain Sistem

a. Kontroler

- Arduino Uno R3 (2 unit)

b. Sensor

- Color Sensor TCS3200
- Load Cell dan HX711 Module

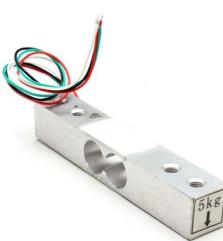
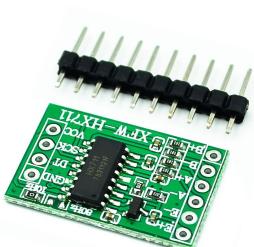
c. Aktuator

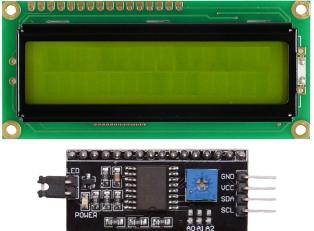
- Micro Servo Motor (2 unit)

d. Komponen pendukung

- LCD 16x2 I2C
- Wire Jumper
- Breadboard

C. Penjelasan Alat dan Komponen yang digunakan

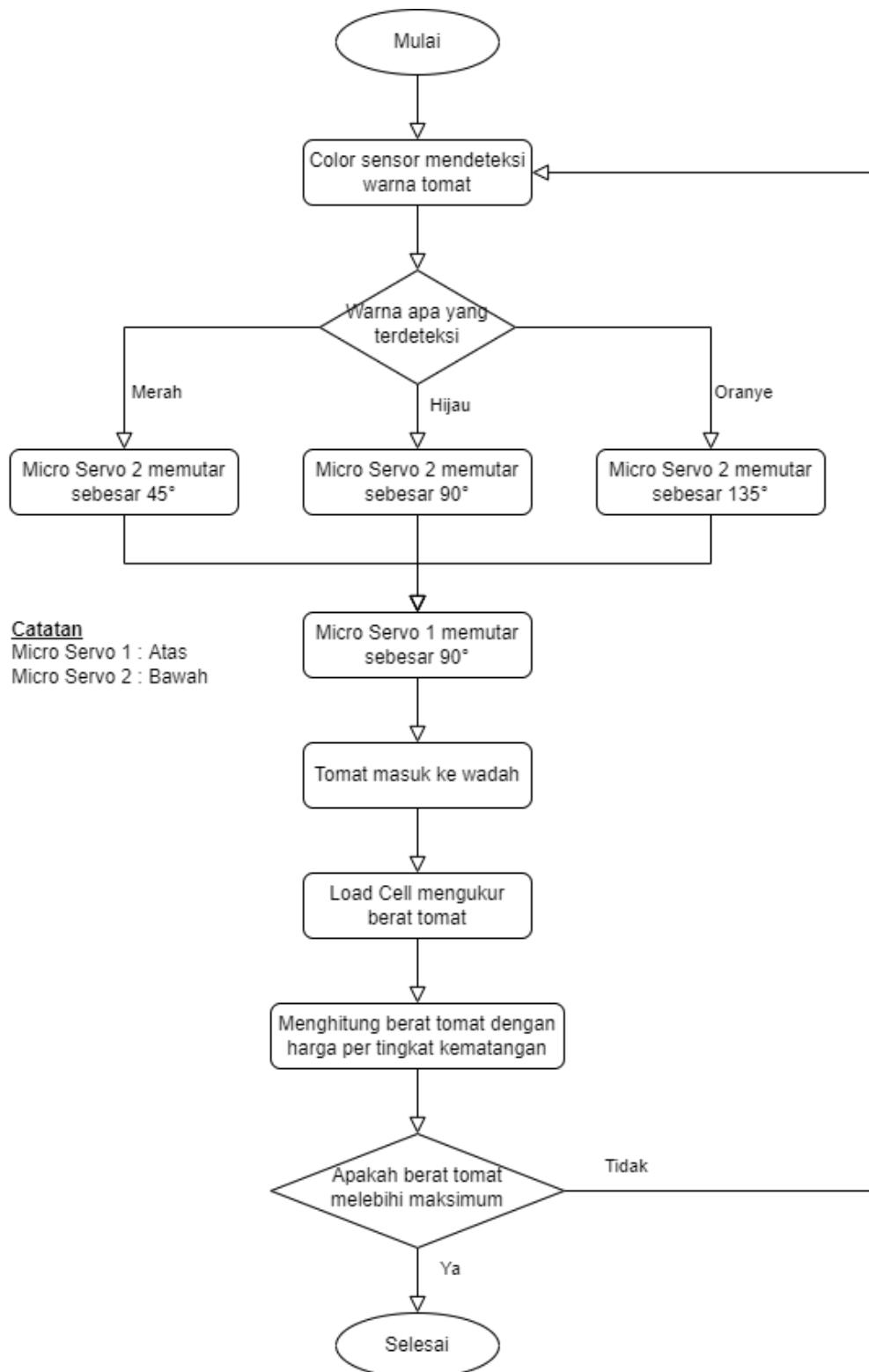
No	Alat	Fungsi Alat
1	Arduino Uno R3 	Arduino berfungsi sebagai pengendali komponen lainnya seperti aktuator, sensor dan lainnya. Terdapat 2 buah Arduino Uno R3 yang digunakan, yang pertama untuk mendeteksi warna dan yang kedua untuk mendeteksi berat dari hasil yang telah disortir.
2	Color Sensor TCS3200 	<i>Color sensor</i> merupakan modul sensor warna yang dapat menampilkan warna dalam format RGB, sehingga pembacaan warna lebih mudah dipahami saat proses pemrogramannya. <i>Color sensor</i> digunakan untuk mendeteksi warna buah tomat dan hasil pembacaannya akan digunakan sebagai masukan arah gerak servo untuk penyortiran.
3	Micro Servo Motor 	Micro servo motor merupakan aktuator putar yang dapat diatur derajat putarnya dan menentukan posisi sudut porosnya. Pada proyek ini menggunakan 2 buah servo untuk memutar lengan arah pada alat.
4	Load Cell 5 kg 	<i>Load cell</i> merupakan sensor yang berfungsi untuk mendeteksi berat berdasarkan tekanan akibat beban yang diberikan.
5	HX711 Module 	HX711 Module adalah modul yang berfungsi untuk membaca berat pada sensor berat tersebut.

6	LCD 16x2 I2C 	LCD berfungsi untuk menampilkan <i>output</i> yang diinginkan. Pada program LCD yang digunakan akan menampilkan berat hasil penimbangan dan harga hasil pengalian setiap gramnya.
7	Breadboard 	Breadboard digunakan untuk membantu membuat rangkaian terhubung tanpa harus melakukan pemasangan permanen dengan solder karena bisa dipasang dan dilepas dengan mudah.
8	Wire Jumper 	<i>Wire jumper</i> adalah kabel yang berfungsi menghubungkan satu alat dengan yang lainnya baik secara langsung antar alat atau dengan menggunakan perantara Breadboard. <i>Wire jumper</i> yang digunakan pada proyek ini berjenis <i>female to male</i> dan <i>male to male</i> .

D. Cara Kerja Sistem

1. Terdapat dua lintasan pada alat yang dirancang. Lintasan pertama merupakan lintasan untuk menuju ke *Color sensor*, sedangkan lintasan kedua sebagai lengan untuk mengarahkan ke kotak sortir.
2. Buah tomat akan menggelinding ke arah papan *Color sensor*.
3. *Color sensor* akan mendeteksi warna buah dan mengkategorikan sebagai “matang”, “setengah matang”, atau “belum matang”.
4. Servo pada lintasan kedua akan berputar ke arah kotak sortir yang sesuai.
5. Papan *Color sensor* akan berputar ke arah lintasan kedua, sehingga posisi buah akan jatuh pada lubang menuju lintasan kedua.
6. Buah tomat menggelinding pada lintasan dua dan masuk ke wadah.
7. *Load cell* akan mendeteksi berat buah tomat.
8. LCD akan menampilkan berat dan harga sesuai dengan yang telah diatur.

Berikut *Flowchart* dari sistem yang dibangun.



Gambar 6 *Flowchart* Sistem

E. Kode Program

- *Color Sensor*

<https://github.com/putrimelanitalb/pstr-kelompok-9/blob/main/ColorSorter.ino>

```
#include <Servo.h>
#define S0 2
#define S1 3
#define S2 4
#define S3 7
#define sensorOut 8

Servo myser1; // Micro servo 1
Servo myser2; // Micro servo 2

// Menyimpan frekuensi warna yang dibaca oleh color sensor
int red = 0;
int green = 0;
int blue = 0;
int benar=0, warna=0;

void setup() {
    myser2.attach(5);
    myser1.attach(6);

    // Set S0-S3 sebagai output
    pinMode(S0, OUTPUT);
    pinMode(S1, OUTPUT);
    pinMode(S2, OUTPUT);
    pinMode(S3, OUTPUT);

    // Set output sensor sebagai input
    pinMode(sensorOut, INPUT);

    // Set Pulse Width scaling menjadi 20%
    digitalWrite(S0,HIGH);
    digitalWrite(S1,LOW);

    // Setup Serial Monitor
    Serial.begin(9600);
}

void loop() {
    myser1.write(90); // Micro servo 1 menulis menggeser lengan ke 90
    delay(3000);

    while(benar==0) {
        detectcolor(); // Deteksi warna
        posservo(); // Posisi servo
        delay(3000);
    }

    myser1.write(60);
    delay(500);
    myser1.write(30);
    delay(500);
    myser1.write(0);
    delay(1000);
}
```

```

myser1.write(30);
delay(500);
myser1.write(60);
delay(500);

benar=0;
}

// Fungsi detectcolor
void detectcolor() {
    // Set sensor untuk baca RED saja
    digitalWrite(S2,LOW);
    digitalWrite(S3,LOW);
    // Read output Pulse Width
    red = pulseIn(sensorOut, LOW);
    // Print output
    Serial.print("R = ");
    Serial.print(red);
    delay(100);

    // Set sensor untuk baca GREEN saja
    digitalWrite(S2,HIGH);
    digitalWrite(S3,HIGH);
    // Read output Pulse Width
    green = pulseIn(sensorOut, LOW);
    // Print output
    Serial.print(" G = ");
    Serial.print(green);
    delay(100);

    // Set sensor untuk baca BLUE saja
    digitalWrite(S2,LOW);
    digitalWrite(S3,HIGH);
    // Read output Pulse Width
    blue = pulseIn(sensorOut, LOW);
    // Print output
    Serial.print(" B = ");
    Serial.println(blue);
    delay(100);

/*
    ripe      --> red(41-51), green(100-167), blue(113-140)
    halfripe  --> red(52-62), green(54-77), blue(100-112)
    unripe    --> red(63-80), green(69-99), blue(67-99)
*/
}

int ripe=0, halfripe=0, unripe=0;

if((41<=red)&&(red<=51)) ripe = ripe + 1;
else if((52<=red) && (red<=62)) halfripe = halfripe + 1;
else if((63<=red) && (red<=80)) unripe = unripe + 1;

if((100<=green) && (green<=167)) ripe = ripe + 1;
else if((54<=green) && (green<=77)) halfripe = halfripe + 1;
else if((69<=green) && (green<=99)) unripe = unripe + 1;

if((113<=blue) && (blue<=140)) ripe = ripe + 1;
else if((100<=blue) && (blue<=112)) halfripe = halfripe + 1;

```

```

else if((67<=blue) && (blue<=99)) unripe = unripe + 1;

// Kesimpulan
if((halfripe>ripe) && (halfripe>unripe)) {
    warna=1;
    Serial.println("Setengah matang");
} else if((unripe>ripe) && (unripe>halfripe)) {
    warna=2;
    Serial.println("Tidak matang");
} else {
    warna=3;
    Serial.println("Matang");
}

void posservo() {
    if(warna==1) { // jika warna oranye (setengah matang)
        myser2.write(135);
        benar=1;
    } else if(warna==2) { // jika warna hijau (tidak matang)
        myser2.write(90);
        benar=1;
    } else if(warna==3) { // jika warna merah (matang)
        myser2.write(45);
        benar=1;
    }
}

```

- *Load Cell: Kalibrasi*

https://github.com/putrimelanitalb/pstr-kelompok-9/blob/main/Kalibrasi_LoadCell.ino

```

#include <HX711.h>
#define DOUT A0
#define CLK A1

HX711 scale(DOUT, CLK);

float calibration_factor = 650;
int gram;

void setup() {
    Serial.begin(9600);
    Serial.println("tekan a,s,d,f untuk menaikan calibration_factor ke
10,100,1000,10000");
    Serial.println("tekan z,x,c,v untuk menurunkan calibration_factor ke
10,100,1000,10000");
    Serial.println("Tekan T untuk Tare");
    scale.set_scale();
    scale.tare();
    long zero_factor = scale.read_average();
    Serial.print("Zero factor: ");
    Serial.println(zero_factor);
    delay(1000);
}

void loop() {

```

```

scale.set_scale(calibration_factor);
gram = scale.get_units(), 4;
Serial.print("Reading: ");
Serial.print(gram);
Serial.print(" Gram");
Serial.print(" calibration_factor: ");
Serial.print(calibration_factor);
Serial.println();

if (Serial.available()) {
    char temp = Serial.read();
    if (temp == '+' || temp == 'a')
        calibration_factor += 0.1;
    else if (temp == '-' || temp == 'z')
        calibration_factor -= 0.1;
    else if (temp == 's')
        calibration_factor += 10;
    else if (temp == 'x')
        calibration_factor -= 10;
    else if (temp == 'd')
        calibration_factor += 100;
    else if (temp == 'c')
        calibration_factor -= 100;
    else if (temp == 'f')
        calibration_factor += 1000;
    else if (temp == 'v')
        calibration_factor -= 1000;
    else if (temp == 't')
        scale.tare();
}
}

```

- *Load Cell: Read Input*

https://github.com/putrimelanitalb/pstr-kelompok-9/blob/main/Read_LoadCell.ino

```

#include <HX711.h>
#include <LiquidCrystal_I2C.h>
#define DOUT A0
#define CLK A1

HX711 scale(DOUT, CLK);
LiquidCrystal_I2C lcd(0x27, 16, 2);

float calibration_factor = 504.10;
int gram, total;
int harga=6000;
int satuan=250;

void setup() {
    Serial.begin(9600);
    lcd.init();
    lcd.backlight();
    scale.set_scale();
    scale.tare();
    lcd.clear();
}

```

```

void loop() {
    scale.set_scale(calibration_factor);
    gram = scale.get_units(), 4;
    total = gram * (harga/satuan);

    lcd.setCursor(0, 0);
    lcd.print("Berat: ");
    lcd.print(gram);
    lcd.setCursor(14, 0);
    lcd.print("gr");

    lcd.setCursor(0, 1);
    lcd.print("Harga: Rp ");
    lcd.print(total);

    Serial.println("-----");
    Serial.print("Berat : ");
    Serial.print(gram);
    Serial.println(" gr");

    Serial.print("Harga : Rp ");
    Serial.println(total);
}

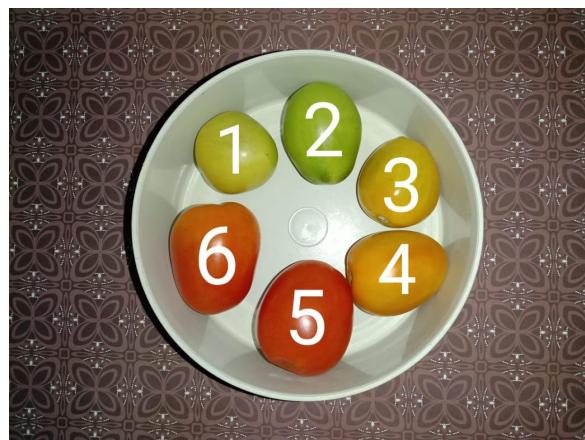
```

V. Pengujian Sistem

A. Skema Pengujian

Pengujian sistem dilakukan menggunakan 6 buah tomat dengan tiga golongan warna yang berbeda. Berikut merupakan detail warna yang digunakan:

- Buah tomat berwarna merah mengindikasikan buah yang matang
- Buah tomat berwarna oranye mengindikasikan buah yang setengah matang
- Buah tomat berwarna hijau mengindikasikan buah yang belum matang



Gambar 7 Buah Tomat yang Digunakan untuk Pengujian

Berikut tingkat kematangan masing-masing buah tomat yang digunakan.

No Buah	Tingkat Kematangan
1	Belum Matang
2	Belum Matang
3	Setengah Matang
4	Setengah Matang
5	Matang
6	Matang

Range nilai RGB yang digunakan untuk mengidentifikasi tingkat kematangan buah tomat didapatkan dari hasil percobaan warna masing-masing buah tomat menggunakan *Color sensor*. Pada percobaan ke-1, nilai RGB diambil secara bergiliran dari Red, Green, dan Blue terhadap masing-masing buah tomat. Sedangkan pada percobaan ke-2, nilai RGB diambil secara bersamaan terhadap masing-masing buah tomat. Berikut tabel data hasil percobaan warna buah tomat yang dilakukan.

No Buah	Percobaan ke-1			Percobaan ke-2		
	R (min-max)	G (min-max)	B (min-max)	R (min-max)	G (min-max)	B (min-max)
1	48-65	54-67	71-95	44-57	53-65	67-92
2	48-80	68-92	94-115	56-79	69-86	90-111
3	45-62	54-91	90-112	43-54	66-96	89-129
4	41-55	75-97	89-123	41-52	77-97	91-127
5	50-63	129-167	103-140	51-64	132-163	98-138
6	51-62	111-137	91-138	50-61	124-141	113-129

Berdasarkan tabel di atas, dapat disimpulkan *range* minimal dan maksimal nilai RGB sementara yang didapatkan adalah.

Range Nilai RGB Sementara			
Tingkat Kematangan	R	G	B
Belum Matang	44-80	53-92	67-115
Setengah Matang	41-62	54-97	89-129
Matang	50-64	111-167	91-140

Setelah melakukan pengujian warna tingkat kematangan dengan menggunakan *range* nilai RGB di atas, masih diperlukan beberapa penyesuaian nilai sehingga mendapatkan *range* nilai RGB yang sudah ditetapkan. Berikut *range* nilai RGB akhir yang akan digunakan pada tahap pengujian selanjutnya.

<i>Range</i> Nilai RGB Tetap			
Tingkat Kematangan	R	G	B
Belum Matang	63-80	69-99	67-99
Setengah Matang	52-62	54-77	100-112
Matang	41-51	100-167	113-140

Pada ujung lintasan akan diletakkan 3 wadah, dengan posisi dari kiri ke kanan, yaitu matang, belum matang, dan setengah matang. Lintasan akan bergerak sesuai dengan buah tomat dengan penggolongan warna sesuai hasil pendektsian warna oleh *Color sensor*. LCD akan menampilkan total berat keseluruhan dari masing-masing wadah. Total harga juga akan ditampilkan sesuai hasil perhitungannya. Berikut data berat aktual dari masing-masing tomat yang digunakan.

No Buah	Berat	No Buah	Berat
1	39 gram	4	53 gram
2	46 gram	5	72 gram
3	38 gram	6	63 gram

Skema pengujian terhadap 6 buah tomat ditampilkan pada tabel berikut.

No Uji	Buah	Kematangan	Arah Lintasan	Sudut Servo (°)	LCD
1	2	Belum matang	Tengah	90	Berat: 46 gr Harga: Rp 1104
2	4	Setengah matang	Kanan	135	Berat: 53 gr Harga: Rp 1272
3	6	Matang	Kiri	45	Berat: 63 gr Harga: Rp 1512
4	1 dan 2	Belum matang	Tengah	90	Berat: 85 gr Harga: Rp 2040
5	3 dan 4	Setengah matang	Kanan	135	Berat: 91 gr Harga: Rp 2184
6	5 dan 6	Matang	Kiri	45	Berat: 135 gr Harga: Rp 3240

Keterangan:

- Harga tomat yang digunakan adalah Rp 6000 per 250 gram.

B. Indikator Keberhasilan

1. *Color sensor* berhasil melakukan sensor warna dengan tepat dan ditampilkan pada serial monitor dengan format RGB dan keterangan kualitas kematangan.
2. Micro servo berhasil mengarahkan lintasan dua ke arah wadah sesuai dengan tingkat kematangan dan sudut yang sesuai.
3. *Load cell* berhasil mendeteksi berat dan LCD berhasil menampilkan hasil pembacaan dengan sesuai.
4. *Load cell* berhasil mengkalkulasikan berat dan LCD menampilkan hasil pembacaan dengan sesuai.
5. Jika dilakukan sortir buah dan terdeteksi tingkat kematangan sama dengan sebelumnya maka program berhasil mengkalkulasikan harga dan menampilkan total berat dan harga di LCD.

VI. Hasil dan Pembahasan

A. Implementasi Sistem

Pengembangan sistem dilakukan secara bertahap mulai dari perancangan alat hingga dilakukan proses uji coba terhadap sistem *software* dan *hardware*. Penentuan ide dilakukan dengan melakukan studi literatur dan *brainstorming* bersama anggota kelompok. Setelah penentuan topik, maka dilakukan studi literatur lebih dalam untuk melakukan analisis kebutuhan alat, bahan dan rangkaian yang akan digunakan. Kemudian, dilakukan tahapan merancang *flowchart* sistem untuk melihat proses kerja sistem.

Implementasi ide selanjutnya dilakukan perancangan diagram alat menggunakan Fritzing dan desain 3D menggunakan Tinkercad. Pada saat perancangan diagram, beberapa modul alat tidak tersedia pada Fritzing, sehingga dilakukan pencarian dan proses instalasi pada fritzing.



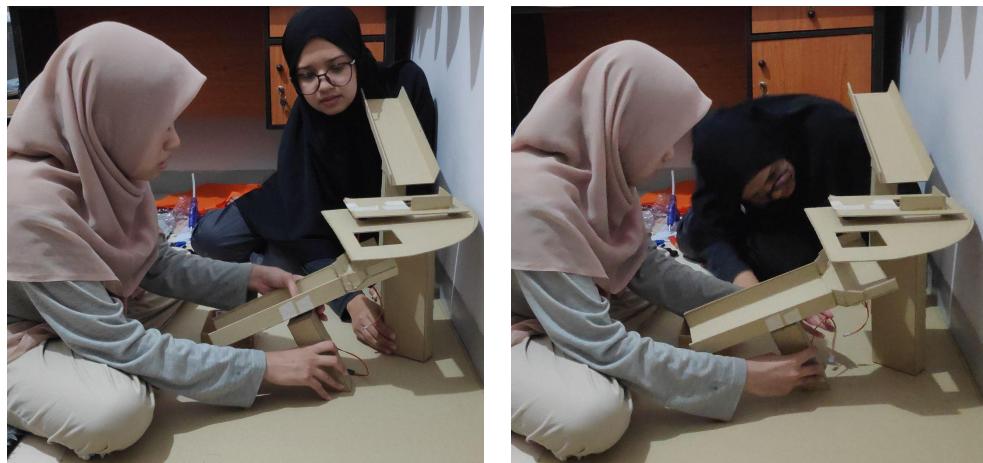
Gambar 8 Proses Perancangan *Hardware*

Hasil desain sistem dan rancangan diagram telah memberikan gambaran yang jelas, sehingga dapat dilakukan perancangan kode program pada Arduino IDE. Proses koding juga disertai pengerjaan perangkaian alat dengan membagi tugas yang setara antara anggota kelompok.



Gambar 9 Perancangan Kode Program *Load Cell*

Pada tahapan perancangan kode program *Load cell* harus diawali dengan proses kalibrasi untuk mencocokkan proses penghitungan berat. Setelah *hardware* selesai dan rangkaian telah dipasang, maka dilakukan proses pengujian dengan beberapa skema yang telah ditentukan sebelumnya. Beberapa permasalahan masih ditemukan sehingga perlu dilakukan penyesuaian dan perbaikan terhadap *hardware* maupun *software*-nya hingga hasil pengujian sesuai dengan yang diharapkan.



Gambar 10 Pemasangan dan Perbaikan *Hardware*

B. Hasil dan Pengujian

Berikut *confusion matrix* hasil pengujian sortir buah tomat menggunakan *Color sensor*.

		Aktual		
		Matang	Belum matang	Setengah matang
Prediksi	Matang	3	1	1
	Belum matang	0	2	0
	Setengah matang	0	0	2

Evaluasi dari hasil pengujian menggunakan persamaan sebagai berikut.

$$\text{Accuracy} = \frac{(TP+TN)}{(TP+TN+FP+FN)}$$

$$\text{Precision} = \frac{TP}{(TP+FP)}$$

$$\text{Recall} = \frac{TP}{(TP+FN)}$$

Keterangan:

TP = *True Positive*

TN = *True Negative*

FP = *False Positive*

FN = *False Negative*

Berikut tabel evaluasi dari hasil *confusion matrix* menggunakan persamaan di atas.

Tingkat Kematangan	Accuracy	Precision	Recall
Matang	77,78%	60%	100%
Belum Matang	88,89%	100%	66,67%
Setengah Matang	88,89%	100%	66,67%

Berdasarkan tabel tersebut, maka dapat disimpulkan bahwa nilai *accuracy* dari masing-masing tingkat kematangan, yaitu 77,78% untuk buah tomat yang matang, 88,89% untuk buah tomat yang belum matang, dan 88,89% untuk buah tomat yang setengah matang. Sehingga, nilai akurasi rata-rata untuk mengidentifikasi kematangan buah tomat adalah 85,18%.

Nilai *precision* untuk buah tomat yang matang sebesar 60%, buah tomat yang belum matang sebesar 100%, dan buah tomat yang setengah matang sebesar 100%. Nilai *recall* untuk buah tomat yang matang sebesar 100%, buah tomat yang belum matang sebesar 66,67%, dan buah tomat yang setengah matang sebesar 66,67%.

Berikut tabel hasil pengujian berat buah tomat menggunakan *Load cell*.

Nomor Tomat	Berat Aktual	Berat dari Load cell	Output Akumulasi pada LCD	Catatan
Hasil Sortir: Matang				
6	63 gram	63 gram	Berat: 63 gr Harga: Rp 1512	Total berat keseluruhan aktual adalah 212 gram .
5	72 gram	72 gram	Berat: 135 gr Harga: Rp 3240	
3	38 gram	39 gram	Berat: 174 gr Harga: Rp 4176	
1	39 gram	39 gram	Berat: 213 gr Harga: Rp 5112	
Hasil Sortir: Belum Matang				
2	46 gram	46 gram	Berat: 46 gr Harga: Rp 1104	
Hasil Sortir: Setengah Matang				
4	53 gram	53 gram	Berat: 53 gr Harga: Rp 1272	

Berdasarkan tabel tersebut, berat aktual dan berat dari *Load cell* dari buah tomat memiliki nilai yang sama. Tetapi, nilai akumulasi berat dari *Load cell* untuk tingkat kematangan buah tomat yang matang memiliki perbedaan selisih sebesar 1 gram dengan total berat aktual menggunakan timbangan. Untuk tingkat kematangan buah tomat yang belum matang dan setengah matang memiliki nilai akumulasi berat yang sama dengan timbangan.

Berikut kesesuaian hasil pengujian terhadap indikator keberhasilannya.

No	Indikator Keberhasilan	Hasil			Catatan
		Sesuai	Kurang Sesuai	Tidak Sesuai	
1	<i>Color sensor</i> berhasil melakukan sensor warna dengan tepat dan ditampilkan pada serial monitor dengan format warna RGB dan keterangan kualitas kematangan.		✓		Pada buah tomat yang setengah matang dan belum matang terdeteksi sebagai buah tomat yang matang.
2	Micro servo berhasil mengarahkan lintasan dua ke arah wadah sesuai dengan tingkat kematangan dan sudut yang sesuai.	✓			
3	<i>Load cell</i> berhasil mendeteksi berat dan LCD berhasil menampilkan hasil pembacaan dengan sesuai.	✓			
4	<i>Load cell</i> berhasil mengkalkulasikan berat dan LCD menampilkan hasil pembacaan dengan sesuai.	✓			
5	Jika dilakukan sortir buah dan terdeteksi tingkat kematangan sama dengan sebelumnya maka program berhasil	✓			

	mengkalkulasikan harga dan menampilkan total berat dan harga di LCD.				
--	--	--	--	--	--

C. Kendala

1. Kesulitan menentukan nilai *range* RGB pada buah tomat.
2. Implementasi desain alat ke rangkaian fisik.
3. Beberapa kali melakukan perubahan rancangan desain karena desain yang sebelumnya sulit untuk diaplikasikan.
4. Kurangnya pengalaman dalam merancang suatu alat, sehingga perhitungan dalam merancang alat sedikit kurang tepat.
5. Nilai yang dihasilkan oleh *Load cell* kurang tepat karena merancang alat *Load cell* dengan bahan seadanya.
6. Harga *Load cell* cukup mahal.

VII. Jadwal Pengerjaan Proyek

Aktivitas	Pertemuan Ke-						
	8	9	10	11	12	13	14
Planning: Merancang ide dan <i>brainstorming</i>							
Studi literatur							
Analisis sistem dan merancang <i>flowchart</i>							
<i>Define</i> kebutuhan alat dan bahan							
Merancang diagram alat dan desain 3D di Tinkercad dan Fritzing							
Merancang kode program alat sortir di Arduino IDE							
Merangkai alat / <i>hardware</i>							
Merancang kode program <i>Load cell</i> di Arduino IDE							
Pengujian sistem dan alat							
Perbaikan							
Pembuatan laporan							
Pembuatan video presentasi dan demo							

VIII. Kesimpulan

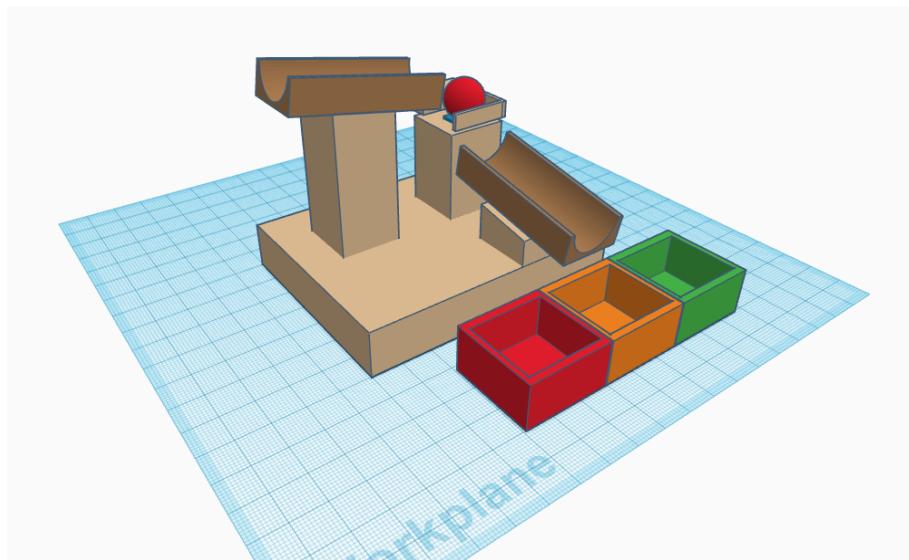
Sistem alat sortir buah tomat yang telah dikembangkan masih belum mampu menyesuaikan warna buah dengan tepat. Hal ini dikarenakan, *channel* warna pada setiap jenis kematangan buah tomat hampir memiliki nilai yang sama. Selain alat sortir buah, pada pengukur berat masih menghasilkan nilai ukuran yang kurang tepat. Oleh karena itu, masih diperlukan perbaikan dan pengembangan sistem yang lebih lanjut.

IX. Saran

Sistem yang telah dirangkai sudah cukup baik. Namun, saat pengujian ditemukan bahwa sensor warna terlalu sensitif terhadap bercak buah dan tidak toleran terhadap kecacatan buah tomat. Sehingga, diperlukan adanya pengaturan rentang toleransi dan untuk meningkatkan kualitas sensor dapat menggunakan sensor warna yang lebih baik. Penggunaan sensor infra merah dapat dicoba untuk mendeteksi warna merah pada buah tomat. Selain itu, penggunaan bahan pembuatan alat dari karton duplex dapat ditingkatkan pada penelitian selanjutnya. Hal ini dilakukan agar alat yang akan dibuat selanjutnya lebih kokoh dan dapat menampung buah yang lebih berat.

X. Daftar Pustaka

- Athifa SF, Rachmat HH. 2019. Evaluasi karakteristik deteksi warna RGB sensor TCS3200 berdasarkan jarak dan dimensi objek. *JETRI: Jurnal Ilmiah Teknik Elektro*. 16(2):105-120. DOI: 10.25105/jetri.v16i2.3459.
- Kusumaningtyas S, Asmara RA. 2016. Identifikasi kematangan buah tomat berdasarkan warna menggunakan metode jaringan syaraf tiruan (JST). *Jurnal Informatika Polinema*. 2(2):72-75. DOI: 10.33795/jip.v2i2.59.
- Simbar RSV, Syahrin A. 2017. Prototype sistem monitoring temperatur menggunakan Arduino Uno R3 dengan komunikasi wireless. *Jurnal Teknologi Elektro*. 8(1):80-86. DOI: 10.22441/jte.v8i1.1381.
- Sopyan BF. 2019. Rancang Bangun Alat Sortir Buah Strawberry Berdasarkan Ukuran Berbasis Internet Of Things (IOT) [skripsi]. Universitas Komputer Indonesia. Tersedia pada: <https://elibrary.unikom.ac.id/id/eprint/890/>.
- Subagyo LA. 2017. Sistem monitoring arus tidak seimbang 3 fasa berbasis Arduino Uno. *Jurnal Teknik Elektro*. 6(3):213-221. Tersedia pada: <https://ejournal.unesa.ac.id/index.php/jurnal-teknik-elektro/article/view/21224>.
- [TAOS] Texas Advanced Optoelectronic Solutions. 2009. *TCS3200 TCS3210 Programmable Color Light-to-Frequency Converter*. TAOS009. United States.
- Wicaksono MF, Hidayat. 2017. *Mudah Belajar Mikrokontroler Arduino*. Bandung: Informatika.

LAMPIRAN

Rangkaian Alat 3D menggunakan Tinkercad



Rangkaian Alat Sortir Buah (*Color Sensor*)



Rangkaian Timbangan (*Load Cell*)