

**KLASIFIKASI BUAH ANGGUR MATANG DAN BUSUK
BERBASIS ARDUINO UNO**



Oleh:

- | | |
|---------------------------------|-----------------------------|
| 1. Dimas Fariz Nabalalah | (215150309111002) |
| 2. Faiz Ibrahim G. | (215150309111005) |
| 3. Satria Wahabi | (215150309111003) |
| 4. Ovriawan Aldo P. P. | (175150301111004) |
| 5. Anggota 5 | (xxxxxxxxxxxxxxxxxx) |
| 6. Anggota 6 | (xxxxxxxxxxxxxxxxxx) |

Dosen Pengampu:

Hurriyatul Fitriyah, S.Kom, M.Kom

**PROGRAM STUDI TEKNIK KOMPUTER
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2021**

Ringkasan

Klasifikasi Buah Anggur Matang dan Busuk Berbasis Arduino UNO adalah alat yang digunakan untuk menentukan buah anggur yang matang berdasarkan berat dan warna buah anggur dengan menggunakan sensor load cell dan TICS3200. Hardware yang digunakan adalah arduino uno, boardband, kabel jumper, load cell, HX711, TICS3200, dan LCD. Metode pengerjaan yang dipakai pada penelitian ini yaitu klasifikasi yang menggunakan naive bayes. Hasil dari klasifikasi yang digunakan pada alat ini menunjukkan bahwa nilai akurasi mencapai 0.93 atau 93%. Di mana hasil tersebut merupakan hasil yang tinggi.

DAFTAR ISI

	Halaman
HALAMAN JUDUL	i
RINGKASAN	ii
DAFTAR ISI.....	iii
DAFTAR GAMBAR.....	iv
DAFTAR TABEL	v
 BAB I BLOK DIAGRAM SISTEM	 1
1.1 Blok Diagram	1
 BAB II PERANGKAT KERAS	 2
2.1 <i>Hardware</i>	2
2.2 Skematik Diagram	2
2.3 Tabel PIN.....	3
2.4 Dokumentasi dan Implementasi Alat.	5
 BAB III DATA	 6
3.1 Data	6
 BAB IV METODE	 9
4.1 <i>Flowchart</i>	9
4.2 Metode dan Algoritma	10
4.2.1 Metode	10
4.2.2 Algoritma.....	10
4.3 Kode Program Arduino UNO.....	10
 BAB V HASIL PENGUJIAN	 16
5.1 Nilai akurasi	16

DAFTAR GAMBAR

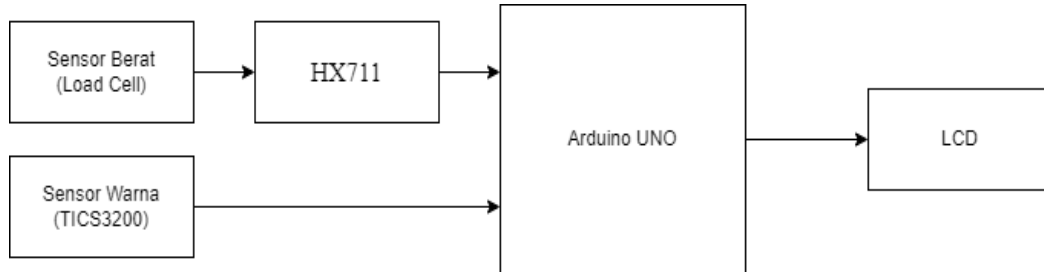
	Halaman
1.1 Blok Diagram	1
2.1 Skema Rangkaian.....	2
2.2 Alat.....	5
4.1 <i>Flowchart</i>	9
5.1 Nilai Akurasi	16

DAFTAR TABEL

	Halaman
2.1 <i>Hardware</i>	2
2.2 TCS3200	3
2.3 HX711	3
2.4 LCD	4
3.1 Data	6

BAB I BLOK DIAGRAM SISTEM

1.1 Blok Diagram



Gambar 1.1 Blok Diagram

Sensor berat (*load cell*) dan sensor warna (TICS3200) adalah input yang digunakan untuk mengklasifikasikan buah anggur yang matang dan busuk. HX711 digunakan sebagai amplifier (penguat sinyal) dan Analog to Digital Converter (ADC) yang berfungsi untuk mengubah sinyal analog menjadi sinyal digital dari sensor load cell. Arduino UNO digunakan sebagai tempat memproses data input sensor berat dan sensor warna. LCD digunakan untuk menampilkan hasil output buah anggur yang matang dan busuk.

BAB II PERANGKAT KERAS

2.1 Hardware

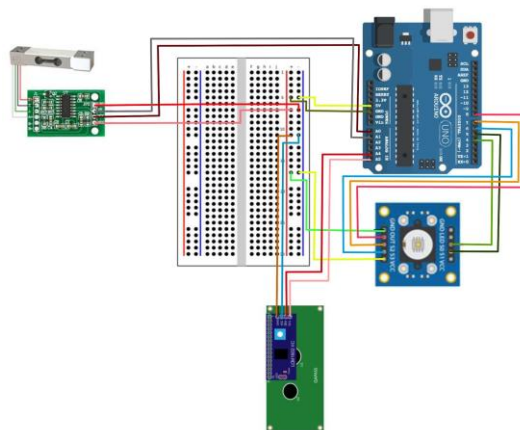
Daftar perangkat keras yang kami gunakan dapat dilihat pada Tabel berikut.

Tabel 2. 1 *Hardware*

No	Nama Komponen	Keterangan
1	Arduino	Untuk memproses kerja alat
2	Sensor Warna Tics3200	Untuk mendeteksi warna
3	Sensor Berat HX711	Untuk mendeteksi berat
4	LCD	Untuk menampilkan hasil
5	Breadboard	Untuk merangkai alat
6	Kabel Jumper	Untuk menyambungkan komponen

2.2 Skematik Diagram

Skematik rangkaian dari alat yang telah dibuat dapat dilihat pada Gambar 2.1.



Gambar 2.1 Skema Rangkaian

2.3 Tabel PIN

Berikut merupakan daftar PIN yang digunakan pada masing-masing komponen. Untuk daftar PIN dari sensor TICS3200 (Sensor Warna) dapat dilihat pada Tabel 2.2 Untuk daftar PIN dari sensor HX711 (Sensor Berat) dapat dilihat pada Tabel 2.3. Untuk daftar PIN dari sensor LCD (16x2) dapat dilihat pada Tabel 2.4.

Tabel 2.2 TCS3200

TICS3200	Arduino
VCC (Input Daya)	5V
GND	GND
S0	PIN 4
S1	PIN 5
S2	PIN 6
S3	PIN 7
OUT	PIN 8

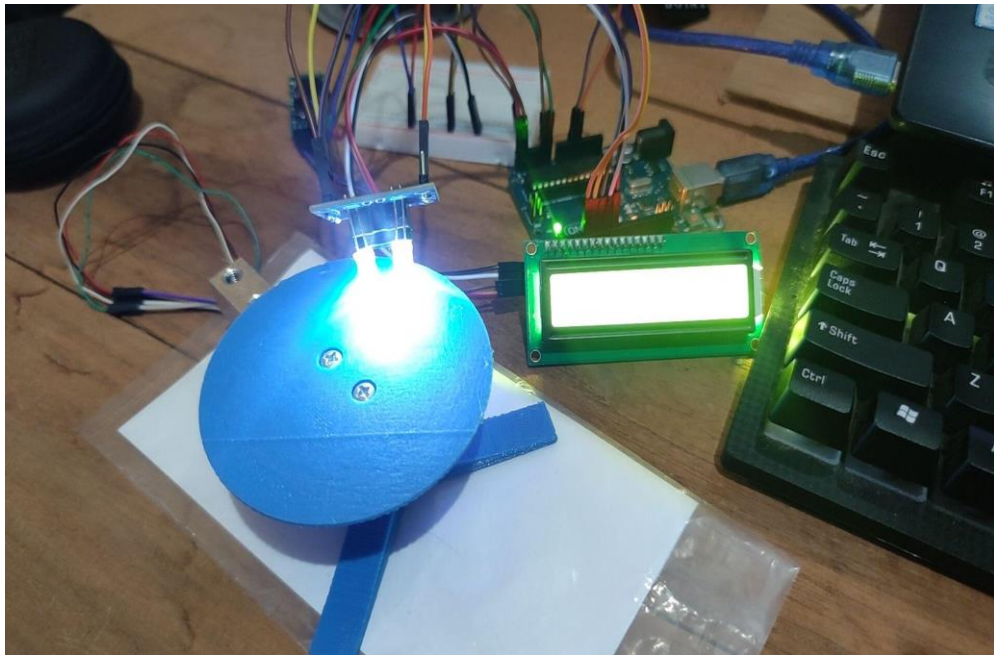
Tabel 2. 3 HX711

HX711	Arduino
VCC	5V
GND	GND
SCK	A0
DT	A1

Tabel 2. 4 LCD

LCD	Arduino
VCC	5V
GND	GND
SDA	A4
SCL	A5

2.4 Dokumentasi dan Implementasi Alat.



Gambar 2.2 Alat

BAB III DATA

3.1 Data

Berikut merupakan data dari pengukuran buah anggur merah yang telah dilakukan secara langsung menggunakan sensor TICS3200 (Warna) dan HX711 (Berat) sebanyak lima puluh kali dan disertakan pada Tabel berikut.

Tabel 3. 1 Data

	Red	Berat	
No	Variabel A (Red)	Variabel B	Kelas
1	75	10	Matang
2	72	6	Matang
3	76	6	Matang
4	79	10	Matang
5	96	9	Matang
6	83	10	Matang
7	67	10	Matang
8	92	7	Matang
9	94	8	Matang
10	81	9	Matang
11	69	8	Matang
12	71	9	Matang
13	70	6	Matang
14	71	8	Matang
15	87	9	Matang
16	72	9	Matang
17	83	10	Matang
18	76	9	Matang

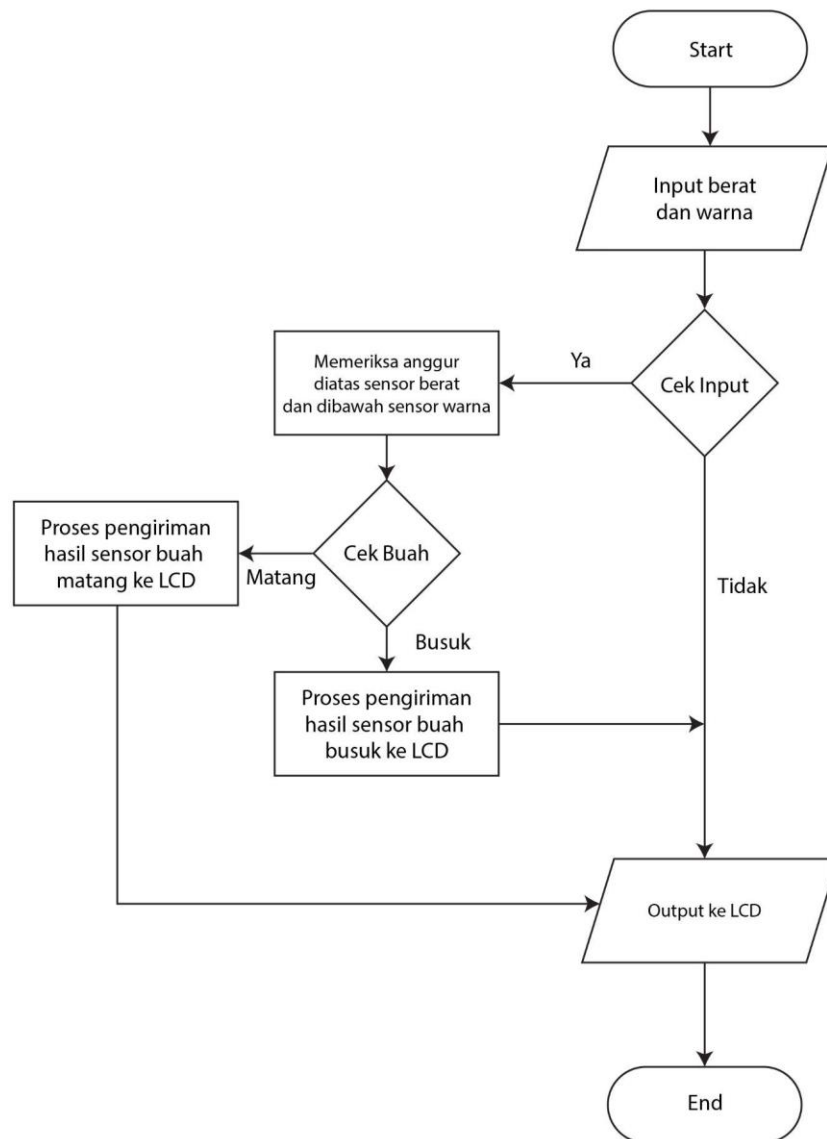
19	89	7	Matang
20	67	9	Matang
21	81	9	Matang
22	80	8	Matang
23	84	7	Matang
24	88	8	Matang
25	80	9	Matang
26	89	7	Busuk
27	79	5	Busuk
28	81	4	Busuk
29	97	6	Busuk
30	95	8	Busuk
31	82	8	Busuk
32	79	5	Busuk
33	86	4	Busuk
34	93	4	Busuk
35	90	10	Busuk
36	90	8	Busuk
37	95	10	Busuk
38	90	7	Busuk
39	85	5	Busuk
40	86	8	Busuk
41	97	7	Busuk
42	109	5	Busuk
43	92	10	Busuk
44	84	8	Busuk

45	75	4	Busuk
46	95	5	Busuk
47	99	5	Busuk
48	108	5	Busuk
49	80	3	Busuk
50	63	5	Busuk

Dari pengukuran warna dan berat pada 50 buah anggur di atas maka dapat diambil datanya bahwa buah anggur pada kelas matang mempunyai rentang indeks warna dari yang terendah 63 sampai yang tertinggi 96 dan rentang indeks berat dari yang terendah 6 sampai yang tertinggi 10. Sementara itu buah anggur pada kelas busuk mempunyai rentang indeks warna dari yang terendah 63 sampai yang tertinggi 109 dan rentang indeks berat dari yang terendah 4 sampai yang tertinggi 10.

BAB IV METODE

4.1 Flowchart



Gambar 4.1 Flowchart

4.2 Metode dan Algoritma

4.2.1 Metode

Pada gambar 4.1 diperlihatkan proses yang terdiri dari proses penentuan buah anggur matang dan buah anggur busuk. Dalam percobaan kali ini kami menggunakan sampel 25 anggur matang dan 25 anggur busuk, yang dimana nantinya akan dibaca oleh 2 sensor yaitu sensor warna (TCS3200) dan sensor berat Loadcell (HX711). Dalam pengambilan sampel warna kami hanya menggunakan sampel warna merah dan untuk berat kami menggunakan loadcell 1 kg.

4.2.2 Algoritma

Mengukur anggur matang dan anggur busuk:

- a. Siapkan anggur mentah dan anggur busuk.
- b. Nyalakan alat dengan memasang ke sumber listrik.
- c. Tunggu hingga alat siap digunakan.
- d. Letakkan 1 biji buah anggur di atas sensor berat dan sensor warna.
- e. Tunggu hingga sensor menghitung dan menampilkan ke LCD.
- f. Ketika hasilnya tampil maka catat data yang telah diambil.

4.3 Kode Program Arduino UNO

Berikut adalah penjelasan dari kode program arduino:

1. Kode program dibawah ini menjelaskan tentang pengambilan library dan pendefinisian pin. HX711_ADC.h untuk library loadcell, Untuk pendefinisian pin *dout* kedalam A1 dan pin *sck* kedalam A0. LiquidCrystal_I2C.h untuk tampilan LCD yang sudah menggunakan modul I2C, dimana kami menggunakan LCD 16x2 untuk projek kali ini dengan alamat 0x27. EEPROM.h digunakan untuk memanggil atau menyimpan hasil kalibrasi. Lalu pendefinisian dari S0,S1,S2,S3 yaitu pin dari TCS3200. Lalu integer disini digunakan untuk pembuatan variabel yang nanti akan digunakan untuk memanggil hasil.

```
//#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include<HX711_ADC.h>
#include<EEPROM.h>
#include<LiquidCrystal_I2C.h>
```

```

#define S0 4
#define S1 5
#define S2 6
#define S3 7
#define sensorOut 8
LiquidCrystal_I2C lcd(0x27,16,2);
const int HX711_dout = A1;
const int HX711_sck = A0;
HX711_ADC LoadCell(HX711_dout, HX711_sck);
const int calVal_eepromAddress = 0;
long t;
int redFrequency = 0;
int redMatang = 0;
int BeratMatang = 0;
int redBusuk = 0;
int BeratBusuk = 0;

```

2. Selanjutnya kode program didalam void setup yang digunakan untuk mendeklarasikan variable dimana untuk HX711_dout sebagai input dari Loadcell kedalam arduino. Lalu pendeklarasi dari S0,S1,S2,S3 yaitu pin output dari TCS3200. Setelah itu pada bagian EEPROM.get berfungsi sebagai mengambil hasil kalibrasi yang telah dilakukan sebelumnya dan memastikan antara loadcell dengan modul HX711 sudah terpasang dengan baik.

```

void setup() {
  Serial.begin(9600);
  lcd.begin();
  pinMode(HX711_dout, INPUT);

  pinMode(S0, OUTPUT);
  pinMode(S1, OUTPUT);
  pinMode(S2, OUTPUT);
  pinMode(S3, OUTPUT);
}

```



```

pinMode(sensorOut, INPUT);
digitalWrite(S0,HIGH);
digitalWrite(S1,HIGH);
delay(10);
Serial.println();
Serial.println("Memulai...");
lcd.begin();
LoadCell.begin();
float calibrationValue;
calibrationValue = 696.0;
EEPROM.get(calVal_eepromAdress, calibrationValue);
long stabilizingtime = 2000;
boolean _tare = true;
LoadCell.start(stabilizingtime, _tare);
if (LoadCell.getTareTimeoutFlag()) {
    Serial.println("Timeout, cek kabel MCU>HX711 pastikan sudah tepat");
    while (1); }
else {
    LoadCell.setCalFactor(calibrationValue);
    Serial.println("Startup selesai"); }
}

```

3. Kode program yang selanjutnya yaitu Void loop yang dimana bertujuan untuk mengeksekusi dan menjalankan program yang sudah dibuat. Untuk memulai penggunaan dari loadcell, loadcell akan mengupdate dari sensor yang telah diberi beban setelah itu jika memang ada maka akan langsung mengeksekusi dan menampilkan ke LCD. Lalu program selanjutnya mendefinisikan dari berat dan warna yang telah di diambil dari sampel. Setelah itu akan dijalankan dengan permisalan jika redMatang lebih dominan dari redBusuk maka hasil yang ditampilkan adalah matang.

```

void loop() {

```

```

//Pemanggilan data loadcell
static boolean newDataReady = 0;
const int serialPrintInterval = 0;

if (LoadCell.update()) newDataReady = true;
if (newDataReady) {
  if (micros() > t + serialPrintInterval) {
    int i = LoadCell.getData();
    if(i<0){
      i=0;
    }
    tampil(i);
    newDataReady = 0;
    t = micros();
  }
}

lcd.setCursor(0,0);
lcd.print("Brt: ");
lcd.setCursor(6,0);
lcd.print("");
lcd.setCursor(7,0);
lcd.print("gr");
lcd.begin();
  lcd.setCursor(0,1);
  lcd.print("Wrn: ");
  lcd.setCursor(6,1);
  lcd.print("");
  redFrequency = pulseIn(sensorOut, LOW);
  lcd.print(redFrequency);
  lcd.print(" ");
  redMatang = map(redFrequency, 67, 96, 255,0);

```

```

BeratMatang = map(HX711_dout, 6, 10, 255,0);
redBusuk = map(redFrequency, 63, 109, 255,0);
BeratBusuk = map(HX711_dout, 4, 10, 255,0);

```

```

if(redMatang > redBusuk ){
    Serial.println("Buah Matang");
    lcd.setCursor(10, 1);
    lcd.print("Matang ");
}

```

```

if(redBusuk > redMatang ){
    Serial.println("Buah Busuk");
    lcd.setCursor(10, 1);
    lcd.print("Busuk ");
}
}

```

4. Untuk dibagian void tampil kami menggunakan ini untuk merapikan pembacaan data dalam loadcell.

```

void tampil(int j){
    //Pendefinisian dari loadcell ke lcd
    lcd.setCursor(4,0);
    lcd.print("  ");
    if(j<10){
        lcd.setCursor(6,0);
    }else if(j<100 && j>=10){
        lcd.setCursor(5,0);
    }else if(j<1000 && j>=100){
        lcd.setCursor(4,0);
    }
}

```

```
}  
lcd.print(j);  
}
```

BAB V HASIL PENGUJIAN

5.1 Nilai akurasi

Dari hasil hitungan python menggunakan klasifikasi naive bayes, didapatkan hasil nilai akurasi sebesar 93% dan memiliki confusion matrix seperti di gambar 5.1. Di mana nilai tersebut sudah termasuk memiliki tingkat keakuratan yang tinggi.

```
... [Naive Bayes Classification]
[[ 8  0  0]
 [ 0 11  1]
 [ 0  1  9]]
```

	precision	recall	f1-score	support
0	1.00	1.00	1.00	8
1	0.92	0.92	0.92	12
2	0.90	0.90	0.90	10
accuracy			0.93	30
macro avg	0.94	0.94	0.94	30
weighted avg	0.93	0.93	0.93	30

Gambar 5.1 Nilai Akurasi