

# **Perancangan *Arduino Uno* pada *Design Mesin Pick and Place Sorting Colour Automation* untuk Meningkatkan Produktivitas**

**Yuwono Izzah Sutita<sup>1\*</sup>, Heri Irwan<sup>2</sup>**

Program Studi S1 Teknik Industri, Universitas Riau Kepulauan<sup>1,2</sup>

[yuwono030400@gmail.com](mailto:yuwono030400@gmail.com)<sup>1</sup>, [hery04@gmail.com](mailto:hery04@gmail.com)<sup>2</sup>

---

## **Informasi Artikel**

### **Riwayat Artikel:**

Disubmit September 10, 2024

Diterima November 26, 2024

Diterbitkan Desember 30, 2024

---

### **Kata Kunci:**

*Arduino Uno*  
*Intergrated Circuit*  
*TCS3200*  
*ATMega328P*  
*Automation*

---

## **ABSTRAK**

*Arduino Uno* adalah sebuah mikrokontroler open-source yang berbasis pada mikrokontroler Atmel AVR. *Arduino Uno* memungkinkan pengguna untuk merancang dan mengembangkan berbagai proyek elektronik dengan mudah. *Arduino Uno* ini menggunakan *Intergrated Circuit (IC)* yaitu mikrokontroler *ATMega328P*. *Arduino Uno* dirancang agar mudah digunakan oleh pemula sekalipun. Tersedia banyak tutorial dan komunitas yang siap membantu. *Arduino Uno* digunakan untuk mengendalikan putaran motor servo Sg90, dan untuk membaca warna pada sensor *TCS3200*. Tujuan dari jurnal ini untuk mengetahui komponen apa saja yang digunakan untuk mengendalikan *Input/Output* dan juga cara penulisan program dari *Arduino Uno* tersebut. Dari hasil Jurnal ini dapat mengetahui komponen apa saja yang digunakan pada Mesin robot *pick and place sorting colour automation* dan untuk pembuatan rancangan mesin. Dari rancangan ini diharapkan dapat mengendalikan Mesin robot *pick and place sorting colour automation* secara bersamaan.

© This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.

---

### **\*Penulis Korespondensi:**

Yuwono Izzah Sutita

Program Studi S1 Teknik Industri

Universitas Riau Kepulauan

Jl. Pahlawan No.99, Bukit Tempayan, Kec. Batu Aji, Kota Batam, Kepulauan Riau

Email: [yuwono030400@gmail.com](mailto:yuwono030400@gmail.com)

---

## **1. PENDAHULUAN**

Seiring dengan kemajuan teknologi, robopat digerakkan dengan program yang dibuat dengan komputer dengan Software dan bahasa Pemrograman, salah satunya yaitu Bahasa C dan di kombinasikan dengan komponen *Hardware* robot. *Hardware* yang digunakan adalah *Arduino* yang dilengkapi dengan Komponen IC Mikrokontroller *Atmega328P* yang berfungsi sebagai otak utama robot dan sebagai penyimpanan program dan mengaplikasikan dengan berbagai bentuk Gerakan, Sinyal, Tampilan, dan lain sebagainya. Robot ini diprogram dengan menggunakan *Arduino*, karena *Arduino* mempunyai fungsi yang banyak dan juga relative lebih murah [6].

Robot ini bekerja dengan cara yaitu, mengambil barang dan memindahkannya ke subjek atau tempat yang kita inginkan. Robot ini juga bisa berjalan karena menggunakan metode mobil (*Smart Car*) sehingga dapat memudahkan memindahkan objek ke subjek dari tempat yang diinginkan [6]. Meningkatkan efisiensi dan produktivitas pada mesin *Pick And Place* berbasis *Arduino* untuk pemilahan warna memerlukan pendekatan multifaset yang melibatkan optimasi perangkat keras

dan perangkat lunak, peningkatan kualitas dan kuantitas sensor, pengaturan waktu yang efisien, dan integrasi sistem yang baik.

## 2. KAJIAN PUSTAKA

### 2.1 Arduino Uno

*Arduino Uno* adalah *board* Mikrokontroler berbasis *ATmega328* dan pada (*datasheet*) Mikrokontroler ini dilengkapi dengan 32 KB *Flash Memory* dan 2 KB SRAM, yang memiliki 14 Pin *Input* dari *Output digital* dimana 6 Pin *Input* tersebut dapat digunakan sebagai *output PWM* dan 6 Pin *Input* pada board ini dapat berfungsi sebagai *output PWM* dan juga menyediakan 6 Pin *Input* analog. Board ini dilengkapi dengan Osilator kristal 16 MHz, koneksi USB, jack daya, header ICSP, serta tombol ulang. Untuk mengoperasikan Mikrokontroler, Anda hanya perlu menghubungkan *Board Arduino Uno* ke komputer melalui kabel USB, atau menggunakan Adaptor AC-DC atau Baterai sebagai sumber daya untuk menjalankannya [2], [4].



Gambar 1. *Arduino Uno*

### 2.2 Relay

*Relay* merupakan komponen untuk menghubungkan atau memutuskan suatu rangkaian listrik dari yang satu ke yang lainnya. *Relay* dapat bekerja secara otomatis dan digerakkan oleh magnet yang dapat dikontrol. Pemilihan *Coil* dan Terminal kontak harus diperhatikan dengan teliti dan seksama karena dapat menyebabkan Percikan Api saat arus tegangan listrik diputuskan atau ialirkan. Jika hal ini tidak dapat diperhatikan, dapat terjadi *Overheat* berlebihan pada Terminal kontak yang pada akhirnya dapat menyebabkan *relay* rusak [4].



Gambar 2. *Relay* dengan 2 Channel

### 2.3 Sensor

Sensor warna *TCS3200* adalah perangkat yang dapat diprogram yang dilengkapi dengan 64 *Photodioda* untuk mendeteksi intensitas cahaya pada berbagai warna objek. Sensor ini juga memiliki filter frekuensi yang berfungsi sebagai Transduser, mengubah arus listrik menjadi sinyal Frekuensi. *TCS3200* mampu membaca empat mode warna, yaitu merah, hijau, biru, dan tidak berwarna (clear), menggunakan ke-64 *Photodioda* tersebut [1], [3].



**Gambar 3.** Sensor TCS3200

#### 2.4 PWM Speed Controller

Pengontrol kecepatan PWM (*Pulse Width Modulation*) banyak digunakan untuk mengendalikan kecepatan motor DC, serta perangkat lain seperti kipas dan lampu. PWM memungkinkan Anda untuk menyesuaikan daya rata-rata yang disalurkan ke motor dengan memvariasikan siklus kerja sinyal digital. Metode ini efisien dan efektif untuk pengendalian kecepatan [5].



**Gambar 4.** PWM Speed Controller

#### 2.5 Servo Sg90

Servo Sg90 (*Alternating Current*) pada *Arduino* adalah sistem Kontrol yang digunakan untuk mengendalikan servo motor menggunakan papan mikrokontroler *Arduino*. Servo Sg90 Servo ini sangat populer karena ukurannya yang kecil, harga yang terjangkau, dan kemampuannya untuk memberikan kontrol presisi pada sudut rotasi. Servo Sg90 dapat diatur untuk bergerak ke sudut tertentu dengan presisi. Pada umumnya, servo ini memiliki rentang sudut antara 0 hingga 180 derajat. Dengan menggunakan *Arduino*, kamu dapat mengendalikan posisi servo dengan mengirimkan sinyal PWM (*Pulse Width Modulation*) yang sesuai. Servo Sg90 sering digunakan dalam proyek robotika, seperti robot lengan, robot mobil, atau robot yang memerlukan bagian yang dapat bergerak atau berputar pada sudut tertentu. Servo ini memungkinkan Kontrol yang tepat terhadap posisi bagian - bagian robot. Servo Sg90 sering digunakan dalam model-model seperti model pesawat terbang, mobil, atau kapal. Dalam proyek - proyek model ini, servo dapat digunakan untuk mengendalikan bagian-bagian yang bergerak, seperti lengan sayap, roda kemudi, atau *rudder*.

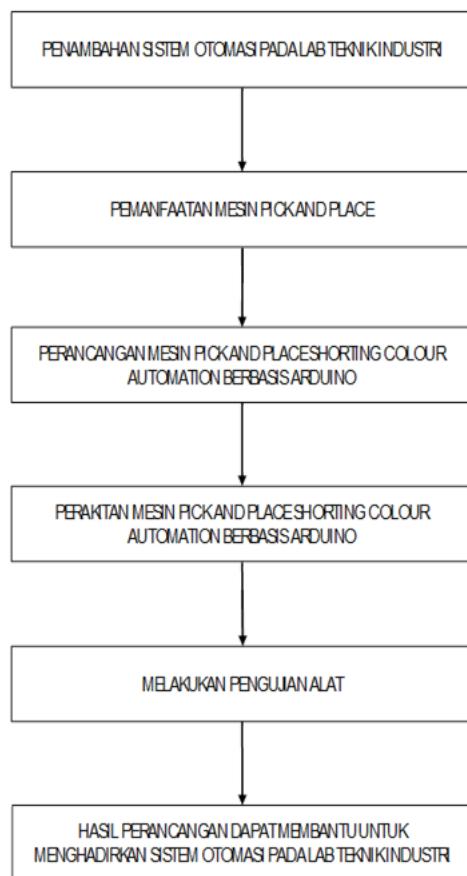


**Gambar 5.** Servo Sg90

### 3. METODE PENELITIAN

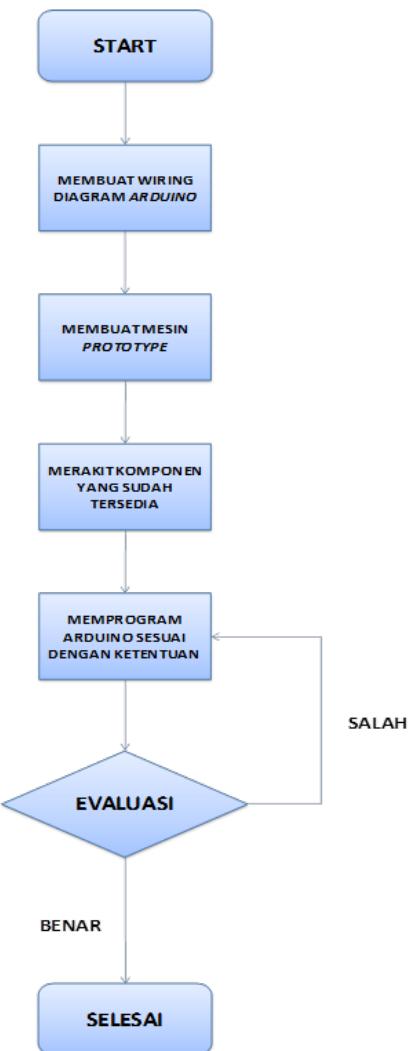
#### 3.1 Flow Chart

Flowchart atau sering disebut dengan diagram alir merupakan suatu jenis diagram yang merepresentasikan algoritma berurutan dalam sistem. Dalam melaksanaan penelitian perlu adanya upaya serangkaian susunan yang harus dilalui agar penelitian bisa berjalan dengan baik dan lebih berfokus pada bagian permasalahan yang akan diteliti , maka dari itu di setiap tahapan - tahapannya saling menentukan tahapan yang lain. Objek dari penelitian ini adalah pembuatan Rancangan *Arduino Uno* Pada *Design Mesin Pick And Place Sorting Colour Automation Berbasis Arduino* Untuk Meningkatkan Produktivitas.



**Gambar 6.** Flowchart Object Penelitian

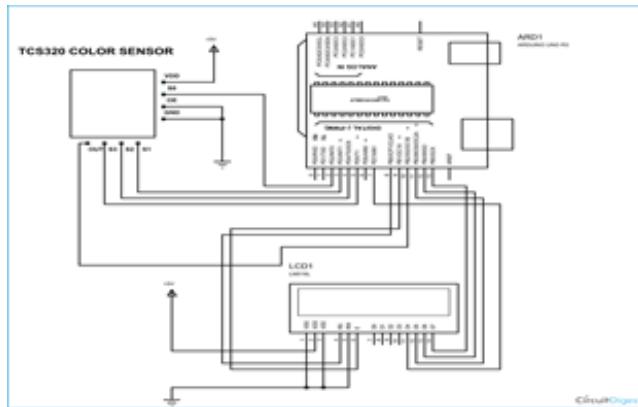
Selanjutnya kita harus membuat susunan panduan untuk mengerjakan Penggeraan *Arduino* berupa *Flowchart*. Dengan menggunakan *flowchart*, pengguna yang mungkin tidak terlalu familiar dengan kode pemrograman agar dapat lebih mudah mengikuti logikanya dari proyek. *Flowchart* menyederhanakan informasi yang kompleks menjadi format yang lebih mudah dimengerti dan dipahami.



**Gambar 7.** *Flowchart* Penggeraan Program *Arduino*

### 3.2 *Wiring Diagram*

*Wiring diagram* adalah representasi grafis dari cara komponen Listrik atau Elektronik dihubungkan dalam suatu lingkaran sirkuit atau sistem. Diagram ini menggambarkan hubungan-hubungan antar berbagai komponen dan menunjukkan bagaimana kabel atau kawat terhubung dari satu bagian ke bagian lainnya. *Wiring diagram* adalah peran penting dalam merancang, membangun, dan memperbaiki sistem Listrik atau Elektronik.



**Gambar 8. Wiring Diagram**

### 3.3 Prototype

Setelah membuat *Wiring Diagram* lalu mengerjakan *Prototype* dengan Skala kecil. *Prototype* adalah model pembuatan simulasi awal atau versi percobaan dari suatu produk, sistem, atau ide - ide yang dibuat untuk tujuan pengujian dan evaluasi. *Prototype* ini difungsikan untuk menguji konsep, desain, dan fungsional sebelum memproduksi versi akhir dari produk. Proses ini sangat penting untuk pengembangan produk, karena dapat membantu untuk mengidentifikasi masalah, mendapatkan *feedback*, dan menyempurnakan hasil desain dari sebelum produksi massal atau implementasi. *Prototype* ini berfungsi sebagai demonstrasi konsep dan memberikan wawasan yang terbaik terkait Desain, Kinerja, dan Fungsi mesin yang diinginkan.

### 3.4 Merakit Komponen

Setelah *Prototype* dibuat dan konsep dari mesin yang kita buat sesuai dengan yang kita ditentukan maka langkah selanjutnya adalah kita akan melakukan perakitan pada komponen - komponen yang akan kita gunakan. Merakit komponen tambahan seperti sensor, Robot Arm dan servo, conveyor atau komponen lainnya yang melibatkan penyambungan pada fisik komponen menggunakan kabel atau mensolder (jika diperlukan). Proses perakitan komponen *Arduino Uno* adalah peran langkah penting dalam pengembangan untuk proyek Elektronik, karena akan memastikan bahwa *Arduino* berfungsi atau berjalan sesuai dengan yang kita buat dan mampu mengendalikan komponen tambahan sesuai dengan program yang kita buat.

### 3.5 Program Arduino

Tahap ini dilakukan jika semua sudah selesai merakit semua Komponen yang terdapat di *Wiring Diagram* yang telah kita buat . Program *Arduino* ini digunakan untuk mengontrol *Input* dan *Output* atau mengendalikan semua komponen Elektronik yang digunakan pada saat mesin yang kita rancang.

```

// sketch_mugSort.ino
// This sketch will read from a color sensor and move servos based on the color
// It uses four servos to move arms to different positions for different colors
// It also includes a LCD1602 display to show the current color being sorted

#include <Servo.h>

Servo servos[4]; // Create four servo objects to control the four arms

void setup() {
  // Initialize the servos
  servos[0].attach(9); // Servo 1 attached to pin 9
  servos[1].attach(10); // Servo 2 attached to pin 10
  servos[2].attach(11); // Servo 3 attached to pin 11
  servos[3].attach(12); // Servo 4 attached to pin 12
}

void loop() {
  // Read the color from the sensor
  int red = analogRead(A0);
  int green = analogRead(A1);
  int blue = analogRead(A2);

  // Set initial arm positions
  servos[0].write(90);
  servos[1].write(90);
  servos[2].write(90);
  servos[3].write(90);

  // Determine the color
  if (red > green & red > blue) {
    servos[0].write(90); // Arm 1 stays at 90 degrees
    servos[1].write(180); // Arm 2 moves to 180 degrees
    servos[2].write(270); // Arm 3 moves to 270 degrees
    servos[3].write(0); // Arm 4 moves to 0 degrees
  } else if (green > red & green > blue) {
    servos[0].write(0);
    servos[1].write(90);
    servos[2].write(180);
    servos[3].write(270);
  } else if (blue > red & blue > green) {
    servos[0].write(180);
    servos[1].write(270);
    servos[2].write(0);
    servos[3].write(90);
  }
}

```

**Gambar 9. Program Arduino Uno**

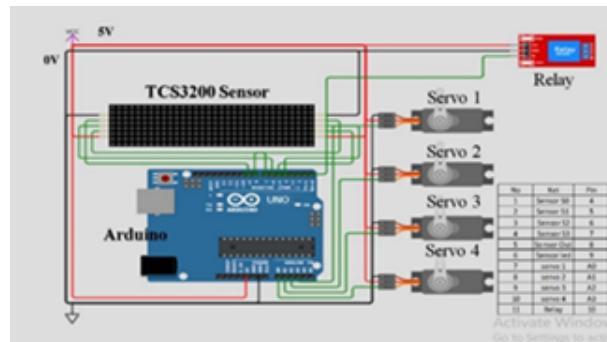
### 3.6 Melakukan Evaluasi

Evaluasi pada program *Arduino* melibatkan serangkaian langkah untuk memastikan bahwa Kode yang ditulis sudah berjalan dengan benar, Efisien, dan sesuai dengan tujuan yang capai . Evaluasi ini mencakup berbagai aspek - aspek, termasuk Pengujian, fungsional, perbaikan Bug, dan Optimasi kinerja. Berikut adalah langkah - langkah umum yang dapat diambil untuk mengevaluasi program *Arduino* secara efektif.

## 4. HASIL DAN ANALISIS

### 4.1 Membuat *Wiring Diagram*

Sebelum melakukan pemrograman *Arduino Uno* kita terlebih dahulu melakukan pembuatan *Wiring Diagram* dari alat Robot *Pick And Place*. Berikut adalah *Wiring Diagram* yang ada pada Robot *Pick And Place*.



Gambar 10. *Wiring Diagram*

Pada gambar di atas tersebut sudah terdapat berberapa komponen *Input dan Output* yang digunakan Seperti Sensor *TCS3200*, *Relay 2 channel*, 4 Motor *Servo* dan *Power Supply*.

### 4.2 Membuat *Prototype*

Selanjutnya kita membuat *Prototype* mini dari mesin Robot *Pick And Place* menggunakan Servo Sg90 yang kecil agar meminimalisir atau mengurangi kesalahan pada proses pemrograman *Arduino Uno* dan juga mengurangi kerusakan komponen Komponen listrik yang akan digunakan. Membuat *prototype* harus menggunakan komponen yang lebih murah dan hemat sebelum menggunakan skala yang Besar.

### 4.3 Merakit Komponen

Proses selanjutnya merakit komponen sesuai dengan *Wiring Diagram* yang telah dibuat. Setelah *Prototype* berhasil dibuat dapat disimpulkan konsep yang kita gunakan agar dapat berjalan dengan baik dan sempurna. Maka langkah selanjutnya adalah kita akan merakit Komponen yang akan kita gunakan pada mesin Robot *Pick And Place*. Sebelumnya kita harus membeli komponen - komponen yang dibutuhkan seperti table dibawah ini.

Tabel 1. Daftar Komponen Mesin Robot *Pick And Place*

No	Nama Komponen	Jumlah	Harga/Pcs	Total Harga
1	Relay 2 Channel	1	Rp 17,000.00	Rp 17,000.00
2	Arduino Uno	1	Rp 60,000.00	Rp 60,000.00
3	Sensor TCS3200	1	Rp 74,000.00	Rp 74,000.00
4	Conveyor	1	Rp 1,075,000.00	Rp 1,075,000.00
5	Robot Arm	1	Rp 220,000.00	Rp 220,000.00
6	Kabel Merah dan	1	Rp 100,000.00	Rp 100,000.00

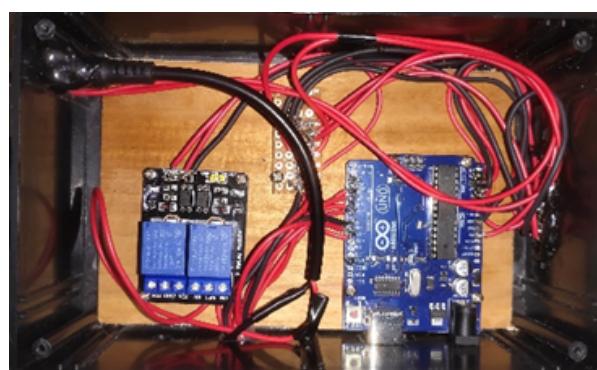
Hitam				
7	PWM Speed Control	1	Rp30,000.00	Rp 30,000.00
8	Box Material 3D Print	3	Rp35,000.00	Rp105,000.00
9	Power Supply 5V	1	Rp 55,000.00	Rp 55,000.00
10	Kayu Papan	10	Rp 40,000.00	Rp 40,000.00
11	Triplek	1	Rp25,000.00	Rp25,000.00
Total			Rp 1,801,000.00	

Pada table diatas menunjukkan daftar komponen yang harus kita dilengkapi dan di kumpulkan seluruhnya beserta harga komponen - komponen tersebut agar kita juga tau biaya pengeluaran yang kita dapat . Dengan jumlah total semua komponen yaitu, Rp. 1.801.000,00. Setelah komponen ini tersedia semua selanjutnya kita merakit penggabungan pada komponen yang satu dengan yang lainnya tersebut sesuai dengan *Input dan Output* yang telah ditentukan. Berikut *Input dan Output* yang digunakan pada mesin Robot *Pick And Place* .

**Tabel 2.** *Input dan Output* yang digunakan

Output	PIN
Sensor S0	4
Sensor S1	5
Sensor S2	6
Sensor S3	7
Sensor Out	8
Sensor LED	9
Servo 1	A0
Servo 2	A1
Servo 3	A2
Servo 4	A3
Relay	10

Pasang dengan teliti dan cocokkan koneksi *Port Pin* yang digunakan pada *Input dan Output* yang digunakan agar Program dapat berjalan dengan baik dan lancar. Selanjutnya rakit semua sampai selesai.



**Gambar 11.** Proses Perakitan *Arduino uno*

#### 4.4 Langkah Membuat Program *Arduino Uno*

Setelah semua perakitan Komponen terhubung dengan baik dan sempurna proses selanjutnya yang kita kerjakan adalah pemograman *Arduino Uno* dengan cara pertama kita menentukan tipe pada data, dan pengontrol yang kita inginkan. Dan dibawah ini merupakan kutipan program yang sudah dibuat dengan menggunakan software *Arduino IDE*, yakni:

1. Menambahkan file Header pada program

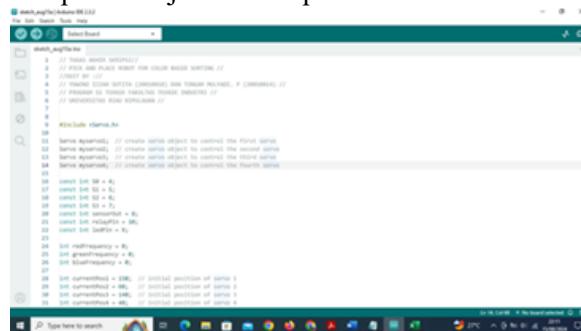
2. Menentukan Pin komunikasi dengan Pin *Arduino*
3. Membuat body program yang akan dilakukan perulangan secara terus menerus selama alat berjalan, pada *Arduino IDE* program tersebut ditulis pada *void loop*.
4. Pengujian setelah pemrograman, dilakukan pengujian keseluruhan untuk memastikan semua komponen bekerja sesuai dengan tugasnya. Jika ditemukan masalah, dilakukan penyesuaian pada program atau pemasangan komponen.

#### 4.5 Kalibrasi dan Penyesuaian Akhir

Setelah pengujian, dilakukan kalibrasi sensor, Robot *Arm*, dan *Conveyor* untuk memastikan akurasi secara operasional. Semua komponen diperiksa kembali untuk menghindari kesalahan dalam proses produksi.

#### 4.6 Pengujian *Hardware*

Pengujian *Hardware* dilakukan agar dapat mengetahui kinerja dari hardware yang digunakan dalam perancangan, pengujian ini terdiri dari pengujian sensor warna. Pada pengujian sensor warna, penulis akan melakukan percobaan dengan mencocokan benda yang berwarna merah, biru dan hitam ke sensor warna. Apakah terjadi *error* apa tidak.



```

// Servo library
#include <Servo.h>

Servo myservo;
int pos = 0;
int pos2 = 90;
int pos3 = 180;
int pos4 = 270;
int pos5 = 360;
int pos6 = 450;
int pos7 = 540;
int pos8 = 630;
int pos9 = 720;
int pos10 = 810;
int pos11 = 900;
int pos12 = 990;
int pos13 = 1080;
int pos14 = 1170;
int pos15 = 1260;
int pos16 = 1350;
int pos17 = 1440;
int pos18 = 1530;
int pos19 = 1620;
int pos20 = 1710;
int pos21 = 1800;
int pos22 = 1890;
int pos23 = 1980;
int pos24 = 2070;
int pos25 = 2160;
int pos26 = 2250;
int pos27 = 2340;
int pos28 = 2430;
int pos29 = 2520;
int pos30 = 2610;
int pos31 = 2700;
int pos32 = 2790;
int pos33 = 2880;
int pos34 = 2970;
int pos35 = 3060;
int pos36 = 3150;
int pos37 = 3240;
int pos38 = 3330;
int pos39 = 3420;
int pos40 = 3510;
int pos41 = 3600;
int pos42 = 3690;
int pos43 = 3780;
int pos44 = 3870;
int pos45 = 3960;
int pos46 = 4050;
int pos47 = 4140;
int pos48 = 4230;
int pos49 = 4320;
int pos50 = 4410;
int pos51 = 4500;
int pos52 = 4590;
int pos53 = 4680;
int pos54 = 4770;
int pos55 = 4860;
int pos56 = 4950;
int pos57 = 5040;
int pos58 = 5130;
int pos59 = 5220;
int pos60 = 5310;
int pos61 = 5400;
int pos62 = 5490;
int pos63 = 5580;
int pos64 = 5670;
int pos65 = 5760;
int pos66 = 5850;
int pos67 = 5940;
int pos68 = 6030;
int pos69 = 6120;
int pos70 = 6210;
int pos71 = 6300;
int pos72 = 6390;
int pos73 = 6480;
int pos74 = 6570;
int pos75 = 6660;
int pos76 = 6750;
int pos77 = 6840;
int pos78 = 6930;
int pos79 = 7020;
int pos80 = 7110;
int pos81 = 7200;
int pos82 = 7290;
int pos83 = 7380;
int pos84 = 7470;
int pos85 = 7560;
int pos86 = 7650;
int pos87 = 7740;
int pos88 = 7830;
int pos89 = 7920;
int pos90 = 8010;
int pos91 = 8100;
int pos92 = 8190;
int pos93 = 8280;
int pos94 = 8370;
int pos95 = 8460;
int pos96 = 8550;
int pos97 = 8640;
int pos98 = 8730;
int pos99 = 8820;
int pos100 = 8910;
int pos101 = 9000;
int pos102 = 9090;
int pos103 = 9180;
int pos104 = 9270;
int pos105 = 9360;
int pos106 = 9450;
int pos107 = 9540;
int pos108 = 9630;
int pos109 = 9720;
int pos110 = 9810;
int pos111 = 9900;
int pos112 = 9990;
int pos113 = 10080;
int pos114 = 10170;
int pos115 = 10260;
int pos116 = 10350;
int pos117 = 10440;
int pos118 = 10530;
int pos119 = 10620;
int pos120 = 10710;
int pos121 = 10800;
int pos122 = 10890;
int pos123 = 10980;
int pos124 = 11070;
int pos125 = 11160;
int pos126 = 11250;
int pos127 = 11340;
int pos128 = 11430;
int pos129 = 11520;
int pos130 = 11610;
int pos131 = 11700;
int pos132 = 11790;
int pos133 = 11880;
int pos134 = 11970;
int pos135 = 12060;
int pos136 = 12150;
int pos137 = 12240;
int pos138 = 12330;
int pos139 = 12420;
int pos140 = 12510;
int pos141 = 12600;
int pos142 = 12690;
int pos143 = 12780;
int pos144 = 12870;
int pos145 = 12960;
int pos146 = 13050;
int pos147 = 13140;
int pos148 = 13230;
int pos149 = 13320;
int pos150 = 13410;
int pos151 = 13500;
int pos152 = 13590;
int pos153 = 13680;
int pos154 = 13770;
int pos155 = 13860;
int pos156 = 13950;
int pos157 = 14040;
int pos158 = 14130;
int pos159 = 14220;
int pos160 = 14310;
int pos161 = 14400;
int pos162 = 14490;
int pos163 = 14580;
int pos164 = 14670;
int pos165 = 14760;
int pos166 = 14850;
int pos167 = 14940;
int pos168 = 15030;
int pos169 = 15120;
int pos170 = 15210;
int pos171 = 15300;
int pos172 = 15390;
int pos173 = 15480;
int pos174 = 15570;
int pos175 = 15660;
int pos176 = 15750;
int pos177 = 15840;
int pos178 = 15930;
int pos179 = 16020;
int pos180 = 16110;
int pos181 = 16200;
int pos182 = 16290;
int pos183 = 16380;
int pos184 = 16470;
int pos185 = 16560;
int pos186 = 16650;
int pos187 = 16740;
int pos188 = 16830;
int pos189 = 16920;
int pos190 = 17010;
int pos191 = 17100;
int pos192 = 17190;
int pos193 = 17280;
int pos194 = 17370;
int pos195 = 17460;
int pos196 = 17550;
int pos197 = 17640;
int pos198 = 17730;
int pos199 = 17820;
int pos200 = 17910;
int pos201 = 18000;
int pos202 = 18090;
int pos203 = 18180;
int pos204 = 18270;
int pos205 = 18360;
int pos206 = 18450;
int pos207 = 18540;
int pos208 = 18630;
int pos209 = 18720;
int pos210 = 18810;
int pos211 = 18900;
int pos212 = 18990;
int pos213 = 19080;
int pos214 = 19170;
int pos215 = 19260;
int pos216 = 19350;
int pos217 = 19440;
int pos218 = 19530;
int pos219 = 19620;
int pos220 = 19710;
int pos221 = 19800;
int pos222 = 19890;
int pos223 = 19980;
int pos224 = 20070;
int pos225 = 20160;
int pos226 = 20250;
int pos227 = 20340;
int pos228 = 20430;
int pos229 = 20520;
int pos230 = 20610;
int pos231 = 20700;
int pos232 = 20790;
int pos233 = 20880;
int pos234 = 20970;
int pos235 = 21060;
int pos236 = 21150;
int pos237 = 21240;
int pos238 = 21330;
int pos239 = 21420;
int pos240 = 21510;
int pos241 = 21600;
int pos242 = 21690;
int pos243 = 21780;
int pos244 = 21870;
int pos245 = 21960;
int pos246 = 22050;
int pos247 = 22140;
int pos248 = 22230;
int pos249 = 22320;
int pos250 = 22410;
int pos251 = 22500;
int pos252 = 22590;
int pos253 = 22680;
int pos254 = 22770;
int pos255 = 22860;
int pos256 = 22950;
int pos257 = 23040;
int pos258 = 23130;
int pos259 = 23220;
int pos260 = 23310;
int pos261 = 23400;
int pos262 = 23490;
int pos263 = 23580;
int pos264 = 23670;
int pos265 = 23760;
int pos266 = 23850;
int pos267 = 23940;
int pos268 = 24030;
int pos269 = 24120;
int pos270 = 24210;
int pos271 = 24300;
int pos272 = 24390;
int pos273 = 24480;
int pos274 = 24570;
int pos275 = 24660;
int pos276 = 24750;
int pos277 = 24840;
int pos278 = 24930;
int pos279 = 25020;
int pos280 = 25110;
int pos281 = 25200;
int pos282 = 25290;
int pos283 = 25380;
int pos284 = 25470;
int pos285 = 25560;
int pos286 = 25650;
int pos287 = 25740;
int pos288 = 25830;
int pos289 = 25920;
int pos290 = 26010;
int pos291 = 26100;
int pos292 = 26190;
int pos293 = 26280;
int pos294 = 26370;
int pos295 = 26460;
int pos296 = 26550;
int pos297 = 26640;
int pos298 = 26730;
int pos299 = 26820;
int pos300 = 26910;
int pos301 = 27000;
int pos302 = 27090;
int pos303 = 27180;
int pos304 = 27270;
int pos305 = 27360;
int pos306 = 27450;
int pos307 = 27540;
int pos308 = 27630;
int pos309 = 27720;
int pos310 = 27810;
int pos311 = 27900;
int pos312 = 27990;
int pos313 = 28080;
int pos314 = 28170;
int pos315 = 28260;
int pos316 = 28350;
int pos317 = 28440;
int pos318 = 28530;
int pos319 = 28620;
int pos320 = 28710;
int pos321 = 28800;
int pos322 = 28890;
int pos323 = 28980;
int pos324 = 29070;
int pos325 = 29160;
int pos326 = 29250;
int pos327 = 29340;
int pos328 = 29430;
int pos329 = 29520;
int pos330 = 29610;
int pos331 = 29700;
int pos332 = 29790;
int pos333 = 29880;
int pos334 = 29970;
int pos335 = 30060;
int pos336 = 30150;
int pos337 = 30240;
int pos338 = 30330;
int pos339 = 30420;
int pos340 = 30510;
int pos341 = 30600;
int pos342 = 30690;
int pos343 = 30780;
int pos344 = 30870;
int pos345 = 30960;
int pos346 = 31050;
int pos347 = 31140;
int pos348 = 31230;
int pos349 = 31320;
int pos350 = 31410;
int pos351 = 31500;
int pos352 = 31590;
int pos353 = 31680;
int pos354 = 31770;
int pos355 = 31860;
int pos356 = 31950;
int pos357 = 32040;
int pos358 = 32130;
int pos359 = 32220;
int pos360 = 32310;
int pos361 = 32400;
int pos362 = 32490;
int pos363 = 32580;
int pos364 = 32670;
int pos365 = 32760;
int pos366 = 32850;
int pos367 = 32940;
int pos368 = 33030;
int pos369 = 33120;
int pos370 = 33210;
int pos371 = 33300;
int pos372 = 33390;
int pos373 = 33480;
int pos374 = 33570;
int pos375 = 33660;
int pos376 = 33750;
int pos377 = 33840;
int pos378 = 33930;
int pos379 = 34020;
int pos380 = 34110;
int pos381 = 34200;
int pos382 = 34290;
int pos383 = 34380;
int pos384 = 34470;
int pos385 = 34560;
int pos386 = 34650;
int pos387 = 34740;
int pos388 = 34830;
int pos389 = 34920;
int pos390 = 35010;
int pos391 = 35100;
int pos392 = 35190;
int pos393 = 35280;
int pos394 = 35370;
int pos395 = 35460;
int pos396 = 35550;
int pos397 = 35640;
int pos398 = 35730;
int pos399 = 35820;
int pos400 = 35910;
int pos401 = 36000;
int pos402 = 36090;
int pos403 = 36180;
int pos404 = 36270;
int pos405 = 36360;
int pos406 = 36450;
int pos407 = 36540;
int pos408 = 36630;
int pos409 = 36720;
int pos410 = 36810;
int pos411 = 36900;
int pos412 = 36990;
int pos413 = 37080;
int pos414 = 37170;
int pos415 = 37260;
int pos416 = 37350;
int pos417 = 37440;
int pos418 = 37530;
int pos419 = 37620;
int pos420 = 37710;
int pos421 = 37800;
int pos422 = 37890;
int pos423 = 37980;
int pos424 = 38070;
int pos425 = 38160;
int pos426 = 38250;
int pos427 = 38340;
int pos428 = 38430;
int pos429 = 38520;
int pos430 = 38610;
int pos431 = 38700;
int pos432 = 38790;
int pos433 = 38880;
int pos434 = 38970;
int pos435 = 39060;
int pos436 = 39150;
int pos437 = 39240;
int pos438 = 39330;
int pos439 = 39420;
int pos440 = 39510;
int pos441 = 39600;
int pos442 = 39690;
int pos443 = 39780;
int pos444 = 39870;
int pos445 = 39960;
int pos446 = 40050;
int pos447 = 40140;
int pos448 = 40230;
int pos449 = 40320;
int pos450 = 40410;
int pos451 = 40500;
int pos452 = 40590;
int pos453 = 40680;
int pos454 = 40770;
int pos455 = 40860;
int pos456 = 40950;
int pos457 = 41040;
int pos458 = 41130;
int pos459 = 41220;
int pos460 = 41310;
int pos461 = 41400;
int pos462 = 41490;
int pos463 = 41580;
int pos464 = 41670;
int pos465 = 41760;
int pos466 = 41850;
int pos467 = 41940;
int pos468 = 42030;
int pos469 = 42120;
int pos470 = 42210;
int pos471 = 42300;
int pos472 = 42390;
int pos473 = 42480;
int pos474 = 42570;
int pos475 = 42660;
int pos476 = 42750;
int pos477 = 42840;
int pos478 = 42930;
int pos479 = 43020;
int pos480 = 43110;
int pos481 = 43200;
int pos482 = 43290;
int pos483 = 43380;
int pos484 = 43470;
int pos485 = 43560;
int pos486 = 43650;
int pos487 = 43740;
int pos488 = 43830;
int pos489 = 43920;
int pos490 = 44010;
int pos491 = 44100;
int pos492 = 44190;
int pos493 = 44280;
int pos494 = 44370;
int pos495 = 44460;
int pos496 = 44550;
int pos497 = 44640;
int pos498 = 44730;
int pos499 = 44820;
int pos500 = 44910;
int pos501 = 45000;
int pos502 = 45090;
int pos503 = 45180;
int pos504 = 45270;
int pos505 = 45360;
int pos506 = 45450;
int pos507 = 45540;
int pos508 = 45630;
int pos509 = 45720;
int pos510 = 45810;
int pos511 = 45900;
int pos512 = 45990;
int pos513 = 46080;
int pos514 = 46170;
int pos515 = 46260;
int pos516 = 46350;
int pos517 = 46440;
int pos518 = 46530;
int pos519 = 46620;
int pos520 = 46710;
int pos521 = 46800;
int pos522 = 46890;
int pos523 = 46980;
int pos524 = 47070;
int pos525 = 47160;
int pos526 = 47250;
int pos527 = 47340;
int pos528 = 47430;
int pos529 = 47520;
int pos530 = 47610;
int pos531 = 47700;
int pos532 = 47790;
int pos533 = 47880;
int pos534 = 47970;
int pos535 = 48060;
int pos536 = 48150;
int pos537 = 48240;
int pos538 = 48330;
int pos539 = 48420;
int pos540 = 48510;
int pos541 = 48600;
int pos542 = 48690;
int pos543 = 48780;
int pos544 = 48870;
int pos545 = 48960;
int pos546 = 49050;
int pos547 = 49140;
int pos548 = 49230;
int pos549 = 49320;
int pos550 = 49410;
int pos551 = 49500;
int pos552 = 49590;
int pos553 = 49680;
int pos554 = 49770;
int pos555 = 49860;
int pos556 = 49950;
int pos557 = 50040;
int pos558 = 50130;
int pos559 = 50220;
int pos560 = 50310;
int pos561 = 50400;
int pos562 = 50490;
int pos563 = 50580;
int pos564 = 50670;
int pos565 = 50760;
int pos566 = 50850;
int pos567 = 50940;
int pos568 = 51030;
int pos569 = 51120;
int pos570 = 51210;
int pos571 = 51300;
int pos572 = 51390;
int pos573 = 51480;
int pos574 = 51570;
int pos575 = 51660;
int pos576 = 51750;
int pos577 = 51840;
int pos578 = 51930;
int pos579 = 52020;
int pos580 = 52110;
int pos581 = 52200;
int pos582 = 52290;
int pos583 = 52380;
int pos584 = 52470;
int pos585 = 52560;
int pos586 = 52650;
int pos587 = 52740;
int pos588 = 52830;
int pos589 = 52920;
int pos590 = 53010;
int pos591 = 53100;
int pos592 = 53190;
int pos593 = 53280;
int pos594 = 53370;
int pos595 = 53460;
int pos596 = 53550;
int pos597 = 53640;
int pos598 = 53730;
int pos599 = 53820;
int pos600 = 53910;
int pos601 = 54000;
int pos602 = 54090;
int pos603 = 54180;
int pos604 = 54270;
int pos605 = 54360;
int pos606 = 54450;
int pos607 = 54540;
int pos608 = 54630;
int pos609 = 54720;
int pos610 = 54810;
int pos611 = 54900;
int pos612 = 54990;
int pos613 = 55080;
int pos614 = 55170;
int pos615 = 55260;
int pos616 = 55350;
int pos617 = 55440;
int pos618 = 55530;
int pos619 = 55620;
int pos620 = 55710;
int pos621 = 55800;
int pos622 = 55890;
int pos623 = 55980;
int pos624 = 56070;
int pos625 = 56160;
int pos626 = 56250;
int pos627 = 56340;
int pos628 = 56430;
int pos629 = 56520;
int pos630 = 56610;
int pos631 = 56700;
int pos632 = 56790;
int pos633 = 56880;
int pos634 = 56970;
int pos635 = 57060;
int pos636 = 57150;
int pos637 = 57240;
int pos638 = 57330;
int pos639 = 57420;
int pos640 = 57510;
int pos641 = 57600;
int pos642 = 57690;
int pos643 = 57780;
int pos644 = 57870;
int pos645 = 57960;
int pos646 = 58050;
int pos647 = 58140;
int pos648 = 58230;
int pos649 = 58320;
int pos650 = 58410;
int pos651 = 58500;
int pos652 = 58590;
int pos653 = 58680;
int pos654 = 58770;
int pos655 = 58860;
int pos656 = 58950;
int pos657 = 59040;
int pos658 = 59130;
int pos659 = 59220;
int pos660 = 59310;
int pos661 = 59400;
int pos662 = 59490;
int pos663 = 59580;
int pos664 = 59670;
int pos665 = 59760;
int pos666 = 59850;
int pos667 = 59940;
int pos668 = 60030;
int pos669 = 60120;
int pos670 = 60210;
int pos671 = 60300;
int pos672 = 60390;
int pos673 = 60480;
int pos674 = 60570;
int pos675 = 60660;
int pos676 = 60750;
int pos677 = 60840;
int pos678 = 60930;
int pos679 = 61020;
int pos680 = 61110;
int pos681 = 61200;
int pos682 = 61290;
int pos683 = 61380;
int pos684 = 61470;
int pos685 = 61560;
int pos686 = 61650;
int pos687 = 61740;
int pos688 = 61830;
int pos689 = 61920;
int pos690 = 62010;
int pos691 = 62100;
int pos692 = 62190;
int pos693 = 62280;
int pos694 = 62370;
int pos695 = 62460;
int pos696 = 62550;
int pos697 = 62640;
int pos698 = 62730;
int pos699 = 62820;
int pos700 = 62910;
int pos701 = 63000;
int pos702 = 63090;
int pos703 = 63180;
int pos704 = 63270;
int pos705 = 63360;
int pos706 = 63450;
int pos707 = 63540;
int pos708 = 63630;
int pos709 = 63720;
int pos710 = 63810;
int pos711 = 63900;
int pos712 = 63990;
int pos713 = 64080;
int pos714 = 64170;
int pos715 = 64260;
int pos716 = 64350;
int pos717 = 64440;
int pos718 = 64530;
int pos719 = 64620;
int pos720 = 64710;
int pos721 = 64800;
int pos722 = 64890;
int pos723 = 64980;
int pos724 = 65070;
int pos725 = 65160;
int pos726 = 65250;
int pos727 = 65340;
int pos728 = 65430;
int pos729 = 65520;
int pos730 = 65610;
int pos731 = 65700;
int pos732 = 65790;
int pos733 = 65880;
int pos734 = 65970;
int pos735 = 66060;
int pos736 = 66150;
int pos737 = 66240;
int pos738 = 66330;
int pos739 = 66420;
int pos740 = 66510;
int pos741 = 66600;
int pos742 = 66690;
int pos743 = 66780;
int pos744 = 66870;
int pos745 = 66960;
int pos746 = 67050;
int pos747 = 67140;
int pos748 = 67230;
int pos749 = 67320;
int pos750 = 67410;
int pos751 = 67500;
int pos752 = 67590;
int pos753 = 67680;
int pos754 = 67770;
int pos755 = 67860;
int pos756 = 67950;
int pos757 = 68040;
int pos758 = 68130;
int pos759 = 68220;
int pos760 = 68310;
int pos761 = 68400;
int pos762 = 68490;
int pos763 = 68580;
int pos764 = 68670;
int pos765 = 68760;
int pos766 = 68850;
int pos767 = 68940;
int pos768 = 69030;
int pos769 = 69120;
int pos770 = 69210;
int pos771 = 69300;
int pos772 = 69390;
int pos773 = 69480;
int pos774 = 69570;
int pos775 = 69660;
int pos776 = 69750;
int pos777 = 69840;
int pos778 = 69930;
int pos779 = 70020;
int pos780 = 70110;
int pos781 = 70200;
int pos782 = 70290;
int pos783 = 70380;
int pos784 = 70470;
int pos785 = 70560;
int pos786 = 70650;
int pos787 = 70740;
int pos788 = 70830;
int pos789 = 70920;
int pos790 = 71010;
int pos791 = 71100;
int pos792 = 71190;
int pos793 = 71280;
int pos794 = 71370;
int pos795 = 71460;
int pos796 = 71550;
int pos797 = 71640;
int pos798 = 71730;
int pos799 = 71820;
int pos800 = 71910;
int pos801 = 72000;
int pos802 = 72090;
int pos803 = 72180;
int pos804 = 72270;
int pos805 = 72360;
int pos806 = 72450;
int pos807 = 72540;
int pos808 = 72630;
int pos809 = 72720;
int pos810 = 72810;
int pos811 = 72900;
int pos812 = 72990;
int pos813 = 73080;
int pos814 = 73170;
int pos815 = 73260;
int pos816 = 73350;
int pos817 = 73440;
int pos818 = 73530;
int pos819 = 73620;
int pos820 = 73710;
int pos821 = 73800;
int pos822 = 73890;
int pos823 = 73980;
int pos824 = 74070;
int pos825 = 74160;
int pos8
```

warna, motor servo atau stepper, dan modul lain yang terhubung ke *Arduino Uno*. Pemeriksaan dilakukan untuk memastikan tidak ada kesalahan wiring yang dapat menyebabkan kerusakan komponen atau gangguan dalam operasi. Evaluasi kode program *Arduino* yang digunakan. Hal ini meliputi penggunaan memori, efisiensi algoritma pengurutan warna, kecepatan eksekusi, dan pengaturan kontrol motor yang tepat.

#### 4.8 Jadwal Kegiatan

Jadwal kegiatan pembuatan mesin *pick and place* adalah sebuah rencana terperinci yang berisi rangkaian aktivitas, durasi, dan urutan pelaksanaan jadwal yang diperlukan untuk menyelesaikan proyek pembuatan mesin tersebut. Jadwal ini sangat penting untuk memastikan proyek berjalan sesuai waktu yang telah ditentukan, meminimalkan risiko keterlambatan, dan untuk mengoptimalkan penggunaan sumber daya. Dibawah ini terdapat tabel Jadwal kegiatan pembuatan mesin *pick and place*.

**Tabel 3.** Jadwal Kegiatan Pembuatan Mesin *pick and place*

NO	TAHAPAN	BULAN												PENANGGUNG JAWAB
		MAR	MAR	MAR	MAR	MAR	MAR	JUNI	JUNI	JULI	JULI	AGUSTUS	AGUSTUS	
1	PENENTUAN CE DAN KONSEP	1												TONGAM, YUDWONO
2	DESAIN (PERANCANGAN)		2	3	4	1	2	3	4	1	2	3	4	TONGAM
3	PEMBELIAN ALAT DAN BAHRAN													TONGAM, YUDWONO
4	PERAKITAN ROBOT ARM													TONGAM
5	PERAKITAN CONVEYOR													YUDWONO
6	PERAKITAN KESELURUHAN													TONGAM, YUDWONO
7	ELECTRICAL													YUDWONO
8	PROGRAMMING													TONGAM, YUDWONO
9	TESTING (PENGUJIAN)													TONGAM, YUDWONO
10	PERBAIKAN DAN PENGEMBANGAN													TONGAM, YUDWONO
11	PEMBUATAN LAPORAN													TONGAM, YUDWONO

#### 4.9 Hasil Akhir

Hasil dari pembuatan mesin *pick and place* untuk sorting berbasis warna menggunakan arduino adalah pengembangan sebuah robot yang mampu secara otomatis mendeteksi, mengambil, dan memindahkan objek berdasarkan warna yang terdeteksi. Sistem ini terdiri dari komponen mekanis seperti lengan robot dan *gripper*, serta komponen elektronik termasuk *Arduino* dan *sensor* warna. Dengan pemrograman yang tepat, robot dapat melakukan tugas sorting dengan akurasi dan kecepatan yang tinggi, menjadikannya prototipe yang berpotensi untuk digunakan dalam aplikasi industri, seperti otomatisasi proses produksi. Hasil pengujian menunjukkan keandalan sistem dalam operasi, dengan kapasitas sorting yang dapat ditingkatkan lebih lanjut.



**Gambar 14.** Design Mesin Robot Pick And Place Sorting Colour Automation Berbasis Arduino Untuk Meningkatkan Produktivitas

## 5. KESIMPULAN DAN SARAN

### 5.1 Kesimpulan

1. Efisiensi: Sistem pick and place sorting warna berbasis Arduino dapat meningkatkan efisiensi proses pemilahan objek berdasarkan warna dibandingkan dengan metode manual. Arduino memungkinkan pengendalian yang akurat dan otomatis dari aktuator dan sensor, yang mempercepat proses sorting.
2. Fleksibilitas: Sistem ini dapat disesuaikan dengan berbagai jenis sensor warna dan aktuator. Ini memungkinkan adaptasi terhadap berbagai kebutuhan industri dan jenis objek yang akan dipilah.

### 5.2 Saran

1. Peningkatan Sensor: Pertimbangkan untuk menggunakan sensor warna yang lebih canggih atau dengan resolusi lebih tinggi untuk meningkatkan akurasi pemilahan warna. Sensor yang lebih baik dapat mengurangi kesalahan akibat variabilitas pencahayaan atau perbedaan warna yang sangat mirip.
2. Kalibrasi: Implementasikan proses kalibrasi yang berkala untuk menjaga akurasi sistem. Kalibrasi yang tepat dapat memastikan bahwa sensor warna tetap akurat seiring waktu dan penggunaan.

## DAFTAR PUSTAKA

- [1] Andrian, Y. (2013). Robot penyortir benda berdasarkan warna menggunakan sensor warna TCS3200. *Sisfotenika*, 3(2), 144-150. [Diakses 24 Februari 2024]
- [2] Almarda, D., & Yusuf, H. (2017). Perancangan Prototype Proteksi Arus Beban Lebih Pada Beban DC Menggunakan Mikrokontroller. *eLEKTUM*, 14(2), 25-34.
- [3] Damayanti, R., Hendrawan, Y., Susilo, B., & Oktavia, S. (2020, April). Prediksi kematangan tomat menggunakan sensor warna TCS3200. Dalam Seri Konferensi IOP: Ilmu Bumi dan Lingkungan (Vol. 475, No. 1, hal. 012011). Penerbitan IOP. DOI 10.1088/1755-1315/475/1/012011 [Diakses 24 Februari 2024]
- [4] Permana, A., Muskita, H. M., Marasabessy, E. W., & Pollatu, F. (2023). Rancang Bangun ATS (Automatic Transfer Switch) Generator Set 3 Phasa Menggunakan Arduino. *Jurnal ELKO (Elektrikal dan Komputer)*, 4(1).
- [5] Rahmadhani, L. (2023). *Rancang Bangun Sistem Pendingin pada Motor Induksi 1 fasa menggunakan kipas PWM berbasis Fuzzy Logic* (Doctoral dissertation, UIN Sunan Gunung Djati Bandung).
- [6] Suwarno, A. (2019). Pengendali Robot Arm Menggunakan Smartphone Android. *Jurnal Gerbang STMIK Bani Saleh*, 9(2). [Diakses 24 Februari 2024]