

# **Panduan Pembuatan Smart Conveyor**



**Rolly Maulana Awangga**  
github : [github.com/awangga](https://github.com/awangga)

**Alit Fajar Kurniawan**  
github : [github.com/alitkurniawan48](https://github.com/alitkurniawan48)

**Informatics Research Center**  
**Applied Bachelor Program of Informatics Engineering**  
**Bandung 2019**

‘Jika Kamu tidak dapat menahan lelahnya belajar,  
Maka kamu harus sanggup menahan perihnya Kebodohan.’  
Imam Syafi’i

## **Acknowledgements**

Pertama-tama kami panjatkan puji dan syukur kepada Allah SWT yang telah memberikan rahmat dan hidayah-Nya sehingga PKPI ini dapat diselesaikan.

## **Abstract**

Panduan Pembuatan Smart Conveyor (PPSC) ini dibuat dengan tujuan memberikan acuan untuk seluruh kalangan agar dapat memahami dan mencoba untuk melakukan pembuatan smart conveyor agar dapat dikembangkan untuk kedepannya. Pada dasarnya pembuatan smart conveyor dibuat dengan tujuan memberikan inovasi terbaru pada perkembangan logistik yang akan terus berkembang mengikuti perkembangan masa. Smart conveyor juga dapat digunakan sebagai alat praktek untuk pembelajaran dibidang logistik.

# Daftar Pustaka

Daftar Pustaka .....	i
Smart Conveyor .....	1
1.1 Deskripsi Smart Conveyor .....	1
1.2 Tujuan.....	1
Pembahasan dan Tutorial .....	2
2.1 Alur Pembahasan .....	2
2.2 Penyiapan Software yang digunakan.....	2
2.3 Merangkai alat SmartConveyor .....	6
2.4 Pemogramman Alat.....	9
Implementasi Alat.....	18
Daftar Pustaka .....	21

# **Smart Conveyor**

Smart Conveyor adalah sebuah prototype yang terangkai dengan sederhana dan digunakan untuk melakukan praktek kerja logistic pada pergudangan [1]. Belt conveyor diterapkan di banyak bidang karena itu ada banyak keuntungan, seperti kapasitas besar, transfer barang mudah dan aman. model prototyping virtual, dan kemudian simulasi dan analisis [2].

## **1.1 Deskripsi Smart Conveyor**

Smart Conveyor pada saat ini sudah sampai di versi 2.0 dan dalam proses menuju versi 3.0. smart conveyor pada versi pertama masih terbentuk sederhana belum adanya pergerakan otomatis terhadap conveyor itu sendiri [3], pada versi 2.0 smart conveyor sudah menggunakan proses otomatis dalam pergerakan conveyor dengan menggunakan sensor ultrasonik. Untuk di versi 3.0 smart conveyor melakukan pengembangan pada proses outbond.

## **1.2 Tujuan**

Berikut tujuan diciptakannya smart conveyor :

1. Memberikan inovasi baru untuk pelaku sector logistik
2. Memberika pelajaran mengenai proses kerja pada pergudangan
3. Dengan menggunakan smart conveyor dapat mengatur peletakan barang secara otomatis

# Pembahasan dan Tutorial

## 2.1 Alur Pembahasan

1. Penyiapan software yang akan digunakan selama tutorial yaitu software Arduino IDE
2. Merangkai alat
3. Pembuatan code pembaca objek dengan menggunakan sensor ultrasonik dan code untuk mengatur keadaan berhenti dan bergerak conveyor
4. Implementasi alat

## 2.2 Penyiapan Software yang digunakan.

### 1.2.1 Instalasi software Arduino IDE

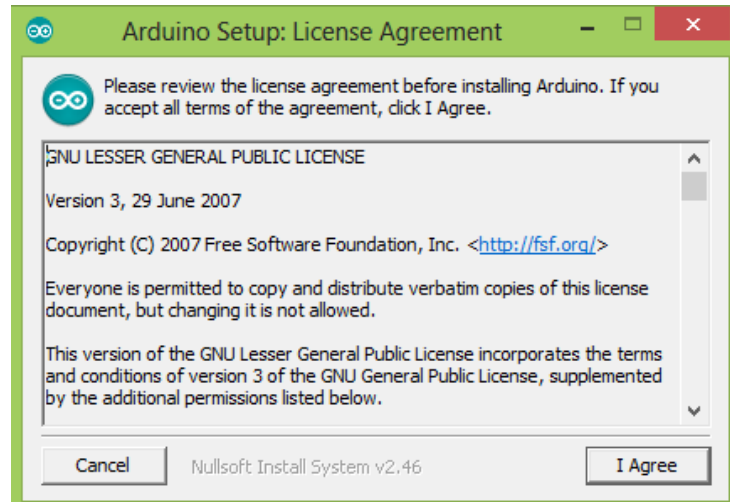
Dalam melakukan instalasi Arduino IDE terutama harus memiliki instaler arduino IDE tersebut, yang dapat di-*search* dan di-download google, atau dapat mengunjungi web ini <https://www.arduino.cc/en/Main/Software>. Kemudian setelah di-download dapat melakukan beberapa tahapan untuk melakukan instalasi.

#### a. PC/Laptop

PC atau Laptop digunakan untuk tempat menginstalasi software arduino.

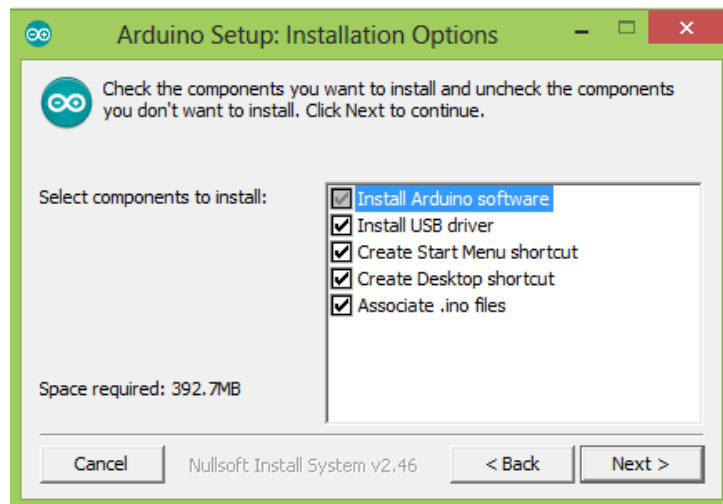
#### b. Arduino IDE

1. Membuka file instalasi Arduino IDE
2. Klik kanan pada *installer* kemudian pilih *Run As Administrator*
3. Masuk ketahapan persetujuan instalasi software arduino IDE. *License Agreement* atau Persetujuan Instalasi, klik tombol *I Agree* untuk memulai install *software*.



**Gambar 1 Proses 1**

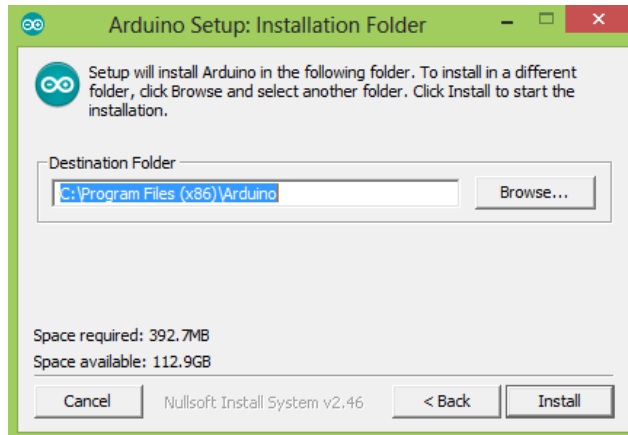
4. Pilih opsi instalasi, Untuk *Installation Option* pilih semua *option* dan klik tombol *Next*.



**Gambar 2 Proses 2**

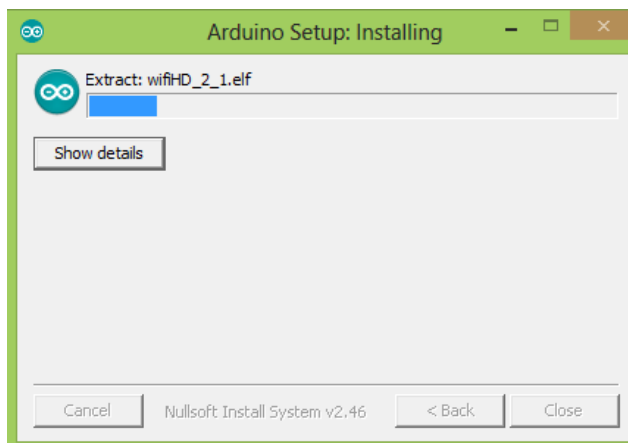
5. Memilih folder, *Installation Folder* atau Pilihan *Folder* untuk memilih *folder* tempat menyimpan program arduino dan klik tombol *install* untuk memulai proses instalasi software, sebaiknya memilih folder yang direkomendasikan.





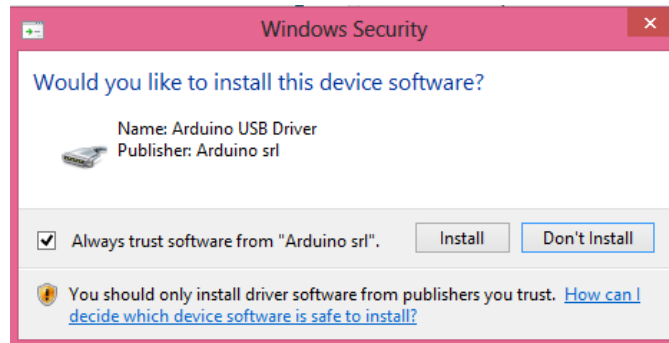
**Gambar 3 Proses 3**

6. Proses instalasi sudah dimulai, tunggu beberapa menit.



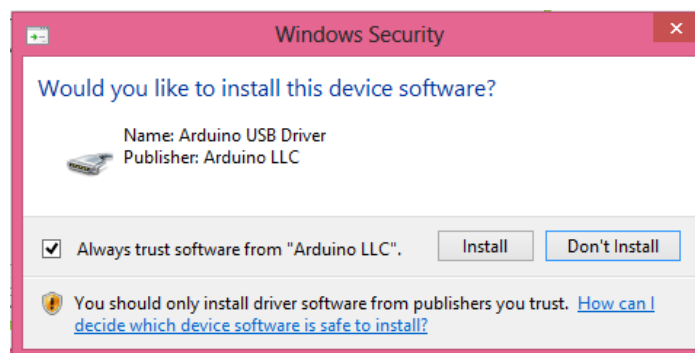
**Gambar 4 Proses 4**

7. Kemudian pada proses instalasi, akan muncul dialog window untuk menginstall Arduino USB Driver, checklist "*Always trust software from Arduini srl*" dan klik *Install*.



**Gambar 5 Proses 5**

8. Kemudian pada proses installasi, akan muncul dialog window ke-dua untuk menginstall Arduino USB Driver, checklist "*Always trust software from Arduino LLC*" dan klik *Install*.

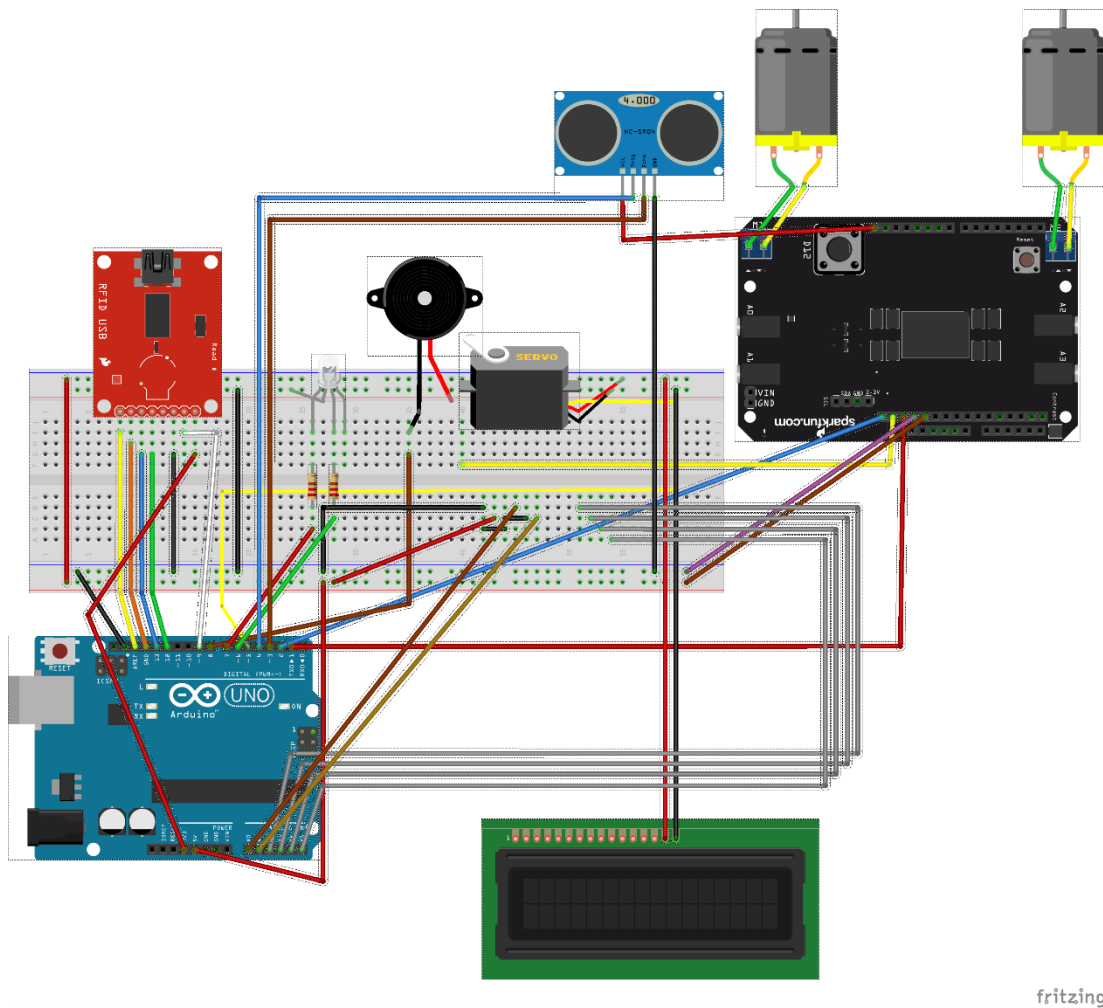


**Gambar 6 Proses 6**

9. Setelah semua proses di atas kamu ikuti, maka Software Arduino IDE telah berhasil terinstall. Klik *Close* untuk menutup dialog window.
10. Kemudian kamu dapat menjalankan software Arduino IDE.

## 2.3 Merangkai alat SmartConveyor

Dalam merangkai alat terlebih dahulu harus mempersiapkan kebutuhan dalam alat dalam merangkai seperti microcontroller Arduino, kanel jumber, RFID reader, servo, breadboard, kabel USB Arduino, LCD, buzzer, led, laptop/pc, sensor ultrasonic, motor DC, module L298n dan software Arduino.



**Gambar 7 Rangkaian alat**

Langkah utama yaitu kita harus mengerti terlebih dahulu fungsi dari setiap pin yang ada pada Arduino, dengan tujuan agar dapat diimplementasikan dalam bentuk code pada software Arduino. Kemudian kita dapat menghubungkan setiap pin Arduino pada berbagai komponen yang digunakan. Cara mnghubungkannya dapat menggunakan kabel jumper sebagai penghubung.

Berikut langkah-langkah pada saat merangkai alat, tahapan pertama merangkai alat

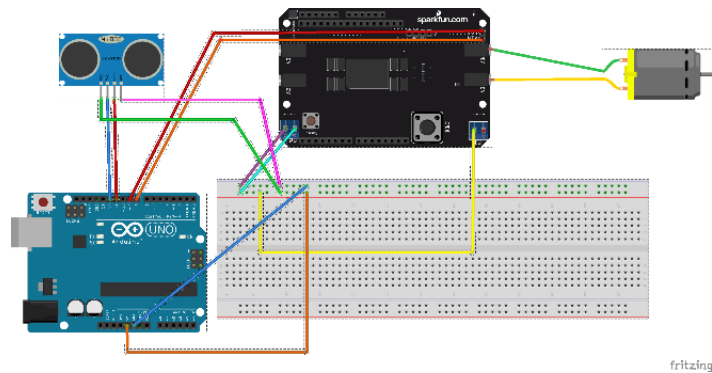
untuk mendeteksi barang dengan RFID.

1. Menghubungkan 5V arduino ke slot negatif pada breadboard, kemudian GND pada arduino ke slot positif breadboard agar sumber daya listrik dapat dibagikan ke banyak komponen sehingga disalurkan ke breadboard.
2. Setiap pin yang ada pada Arduino harus dihubungkan ke breadboard agar dapat digunakan oleh banyak komponen
3. Menghubungkan RFID dengan Arduino dengan menghubungkan GND dan VCC, GND pada pin GND Arduino, dan VCC pada 3volt Arduino. Selebihnya di sesuaikan pada pin digital I/O Arduino (pin 2-13) yang masih kosong.
4. Selanjutnya menghubungkan lcd dengan Arduino, sesuaikan pin GND dan VCC pada lcd dengan Arduino agar lcd mendapatkan energy sumberdaya yang cukup. Pin selebihnya dihubungkan pada pin digital I/O Arduino.
5. Pada proses menghubungkan servo juga sama, harus dapat menyesuaikan VCC dan GND nya. Pin selebihnya dihubungkan pada pin digital I/O Arduino.
6. Pada buzzer dan LED, buzzer LED hanya memiliki 2 alternatif kabel penghubung, yang satu dihubungkan pada GND untuk mendapatkan daya dan kabel penghubung selebihnya pada pin digital I/O
7. Diharapkan setiap pin komponen yang dihubungkan dengan pin digital I/O jangan terlalu banyak menumpuk pada satu pin karena jika banyak terhubung pada satu pin nantinya akan sulit ketika diimplementasikan ke codingan.
8. Setelah semuanya terhubung kemudian sambungkan ke laptop atau PC kemudian upload program yang ada pada Arduino IDE, selanjutnya alat sudah bisa dijalankan.

Selanjutnya tahapan merangkai alat conveyor dengan menggunakan sensor sebagai alat pembawa barang.

1. Pada proses merangkai alat conveyor ini kita menggunakan Arduino yang berbeda, dikarenakan slot pin yang sudah terlalu penuh.
2. Hubungkan arduino dengan sensor dengan menggunakan kabel jumper yaitu menghubungkan Echo Trig pada sensor ultrasonik ke slot 11 dan 12 arduino. Kemudian Vcc sensor ke slot negatif pada breadboard dan GND sensor ke slot positif pada breadboard.
3. Kemudian menghubungkan modul L298N dengan arduino yaitu dengan cara menghubungkan slot input 1 dan 2 pada modul L298N ke slot 9 dan 8 pada arduino.

4. Kemudian untuk mendapatkan tegangan listrik yang stabil maka sambungkan penghubung energi dari modul L298N ke breadboard, yaitu yang 5V ke slot positif pada breadboard, kemudian GND pada modul L298N ke slot negatif yang ada di breadboard dan 12V pada modul dihubungkan ke slot positif pada breadboard. Sehingga energi yang didapatkan dari arduino akan stabil.
5. Selanjutnya yaitu menghubungkan motor DC dengan modul L298N, hubungkan dengan menggunakan kabel dan ikat pada slot output yang berada pada modul L298N.
6. Setelah semuanya terhubung kemudian sambungkan ke laptop atau PC kemudian upload program yang ada pada Arduino IDE, selanjutnya alat sudah bisa dijalankan.



**Gambar 8 Rangkaian Conveyor**

## 2.4 Pemogramman Alat

Setelah melakukan instalasi dan merangkai alat selanjutnya kita melakukan pemogramman arduino nya dengan menggunakan *software* arduino IDE. Langkah awal yaitu hubungkan Arduino Uno dengan PC yang tersedia *software* arduino IDE dengan menggunakan kabel USB arduino. Setelah terhubung maka lampu di arduino uno akan aktif dengan sendirinya.

Tahap selanjutnya yaitu membuka *software* arduino IDE, kemudian melakukan pencocokan *serial port* yang sesuai dengan arduino uno nya tersebut, setelah itu melakukan pemilihan *board* arduino yang digunakan. Karena kita menggunakan arduino IDE maka kita pilih *board* arduino IDE. Kemudian segera melakukan pengkodean untuk dapat berinteraksi dengan mikrokontroler arduino. Berikut *source code* untuk menjalankan alat untuk pendeteksi barang terlenih dahulu.

Setelah membuka software Arduino, kita memulai melakukan pengkodean untuk alat, terlebih dahulu kita harus mengimporkan library yangp digunakan,

```
//import library
#include <deprecated.h>
#include <MFRC522.h>
#include <MFRC522Extended.h>
#include <require_cppll.h>

#include <LiquidCrystal_I2C.h>
#include <Wire.h>

#include <SPI.h>
#include <Ethernet.h>
#include <Servo.h>
```

Kemudian kita melakukan pendefinisian pin digital yang digunakan

```
#define LED_PIN1 A0
#define LED_PIN2 A1
#define LED_PIN3 A2
#define BUZZER_PIN A3

LiquidCrystal_I2C lcd(0x3F, 16, 2);

Servo servo1;
Servo servo2;

String txData = "";

EthernetClient client; //ETHERNET INSTANCE
EthernetClient client2; //ETHERNET INSTANCE

constexpr uint8_t RST_PIN = 9; // Dapat dilakukan konfigurasi, sesuaikan dengan tata letak pin di atas
constexpr uint8_t SS_1_PIN = 4; // mengambil pin yang tidak digunakan sama sekali, untuk mengatur High/Low yang digunakan, berbeda dengan SS_2
constexpr uint8_t SS_2_PIN = 8; // mengambil pin yang tidak digunakan sama sekali, untuk mengatur High/Low yang digunakan, berbeda dengan SS_1

constexpr uint8_t NR_OF_READERS = 2;

byte ssPins[] = {SS_1_PIN, SS_2_PIN};

MFRC522 mfrc522[NR_OF_READERS]; // Membuat instance MFRC522.
```

Selanjutnya kita masuk pada bagian void setup, yang mana Fungsi setup() dipanggil ketika sketsa dimulai. Struktur setup berguna untuk menginisialisasi variabel, mendeklarasikan pin yang digunakan, menggunakan library, dll. Fungsi setup() hanya akan berjalan sekali, setiap menyala atau restart board Arduino.

```
/**
 * Initialize.
 */
void setup() {
  Serial.begin(9600); // Initialize serial communications with the PC
  while (!Serial);    // Do nothing if no serial port is opened (added for Arduinos based on ATMEGA32U4)

  // ETHERNET MODULE INITIAL
  SPI.begin();        // Init SPI bus

  // SERVO
  servol.attach(5);
  servo2.attach(6);

  // LED (GREEN, ORANGE, BLUE)
  pinMode(LED_PIN1, OUTPUT);
  pinMode(LED_PIN2, OUTPUT);
  pinMode(LED_PIN3, OUTPUT);

  // BUZZER
  pinMode(BUZZER_PIN, OUTPUT);

  byte mac[] = {
    0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED
  };    //MAC = 000102030405

  // ETHERNET IP
  byte ip[] = {
    192, 168, 8, 98
  };

  //IPAddress ip(192,168,8,98);
  Ethernet.begin(mac, ip);    //CONNECT USING ABOVE
  Serial.println(Ethernet.localIP());
  Serial.println("SUCCESSFUL CONNECTION!");

  // LCD MODULE INITIAL
  lcd.begin();
  lcd.begin();

  // MUNCULKAN DI LCD
  lcd.backlight();
  lcd.setCursor(4, 0); // BARIS PERTAMA
  lcd.print("WELCOME");
  lcd.setCursor(3, 1); // BARIS KEDUA
  lcd.print("PROFIT-WMS");

  // POSISI DEFAULT SERVO
  servol.write(0);
  servo2.write(0);

  for (uint8_t reader = 0; reader < NR_OF_READERS; reader++) {
    mfrc522[reader].PCD_Init(ssPins[reader], RST_PIN); // Init each MFRC522 card
    Serial.print(F("Reader "));
    Serial.print(reader);
    Serial.print(F(": "));
    mfrc522[reader].PCD_DumpVersionToSerial();
  }
}
```

Setelah kita melakukan penyetingan dengan menginisialisasi dan memberikan nilai awal

pada void setup, selanjutnya kita dapat melanjutkan tahapan pada void loop. Void loop berfungsi untuk melaksanakan atau mengeksekusi perintah program yang telah dibuat. Fungsi loop akan secara aktif mengontrol board Arduino baik membaca input atau merubah output.

```
/**
 * Main loop.
 */
void loop() {

    for (uint8_t reader = 0; reader < NR_OF_READERS; reader++) {
        // Look for new cards

        if (mfrc522[reader].PICC_IsNewCardPresent() && mfrc522[reader].PICC_ReadCardSerial()) {
            Serial.print(F("Reader "));
            Serial.print(reader);
            // Show some details of the PICC (that is: the tag/card)
            Serial.print(F(" : Card UID:"));
            dump_byte_array(mfrc522[reader].uid.uidByte, mfrc522[reader].uid.size);
            Serial.println();
            Serial.print(F("PICC type: "));
            MFRC522::PICC_Type piccType = mfrc522[reader].PICC_GetType(mfrc522[reader].uid.sak);
            Serial.println(mfrc522[reader].PICC_GetTypeName(piccType));

            // DEKLARASI RFID KE VARIABEL
            String strID = "";
            for (byte i = 0; i < 4; i++) {
                strID +=
                    (mfrc522[reader].uid.uidByte[i] < 0x10 ? "0" : "") +
                    String(mfrc522[reader].uid.uidByte[i], HEX) +
                    (i != 3 ? ":" : "");
            }
            strID.toUpperCase();
            strID.replace(":", "");

            // MENKONEKSIKAN KE JARINGAN SERVER DAN MENGIRIM DATA
            //CEK KONEKSI SERVER
            if (client.connect("192.168.8.99", 80) && client2.connect("192.168.8.99", 80)) {
                // MEMANGGIL DATA RFID
                const String ID = strID;
                txData = "barang_id=" + (ID) ;

                // MENGIRIM DATA
                Serial.println("Connected");
                Serial.print(txData);

                client.println("GET /app_wms/inboundc/fetch_data?kode=" + (ID) + "");
                client2.println("GET /app_wms/inboundc/get_jenis?kode=" + (ID));

                //
                client.println("GET /tugas_akhir/index.php?mod=barang_masuk&submod=barang_masuk_check_arduino&barang_id=" + (ID) + "");
                //
                client2.println("GET /tugas_akhir/index.php?mod=barang_masuk&submod=barang_masuk_check_arduino_jenis&barang_id=" + (ID));

                // DELAY 1 DETIK
                delay(500);

                if (client.available() && client2.available())
                {
                    Serial.println("Avaliable");
                    int c = client.read();
                    Serial.println(c);
                    int c2 = client2.read();
                    Serial.println(c2);
                }
            }
        }
    }
}
```



```

        if (c < 49)
        {
            LEDOrange();

            LCDTidakTerdeteksi();

            delay(2000);

            LCDReset();
        } else {
            LEDBlue();

            // MENGERAKKAN SERVO
            if (c2 < 49) {
                LCDPecahBelah();

                ServoSatuGerak();
            } else {
                LCDBukanPecahBelah();

                // ServoDuaGerak();
            }
        }
    } else {
        Serial.println("Not Available");
        LEDBlue();
    }

    // MEMATIKAN KONEKSI
    client.stop();
    client2.stop();
} else {
    Serial.println("Connection Failed.");

    LEDOrange();
}

// Halt PICC
mfrc522[reader].PICC_HaltA();
// Stop encryption on PCD
mfrc522[reader].PCD_StopCryptol();
} //if (mfrc522[reader].PICC_IsNewC
} //for(uint8_t reader
}

/**
 * Helper routine to dump a byte array as hex values to Serial.
 */
void dump_byte_array(byte *buffer, byte bufferSize) {
    for (byte i = 0; i < bufferSize; i++) {
        Serial.print(buffer[i] < 0x10 ? " 0" : " ");
        Serial.print(buffer[i], HEX);
    }
}

```

```

void ServoSatuGerak() {
    servol.write(0);
    delay(15000);
    LCDReset();
    servol.write(90);
}

//void ServoDuaGerak() {
//    servo2.write(0);
//    delay(12000);
//    LCDReset();
//    servo2.write(135);
//}

void LCDReset() {
    lcd.clear();
    lcd.setCursor(4, 0); //baris pertama
    lcd.print("WELCOME");
    lcd.setCursor(3, 1); //baris pertama
    lcd.print("PROFIT-WMS");
}

void LCDPecahBelah() {
    lcd.clear();
    lcd.setCursor(0, 0); //baris pertama
    lcd.print("JENIS BARANG :");
    lcd.setCursor(0, 1); //baris pertama
    lcd.print("PECAH BELAH");
}

void LCDBukanPecahBelah() {
    lcd.clear();
    lcd.setCursor(0, 0); //baris pertama
    lcd.print("JENIS BARANG :");
    lcd.setCursor(0, 1); //baris pertama
    lcd.print("BUKAN PECAH BELAH");
}

void LCDTidakTerdeteksi() {
    lcd.clear();
    lcd.setCursor(0, 0); //baris pertama
    lcd.print("JENIS BARANG :");
    lcd.setCursor(0, 1); //baris pertama
    lcd.print("TIDAK TERDETEKSI");
}

void LEDBlue() {
    // MENGHIDUPKAN LAMPU BIRU DAN BUZZER
    digitalWrite(LED_PIN1, HIGH);
    digitalWrite(LED_PIN2, LOW);
    tone(BUZZER_PIN, 1500);

    // DELAY 0.5 DETIK
    delay(300);

    // MEMATIKAN LAMPU BIRU DAN BUZZER
    digitalWrite(LED_PIN1, LOW);
    noTone(BUZZER_PIN);
}

```

```

void LEDOrange() {
  // MENGHIDUPKAN LAMPU ORANGE DAN BUZZER
  digitalWrite(LED_PIN2, HIGH);
  digitalWrite(LED_PIN1, LOW);
  tone(BUZZER_PIN, 1000);

  // DELAY 1 DETIK
  delay(300);

  // MEMATIKAN LAMPU ORANGE DAN BUZZER
  digitalWrite(LED_PIN2, LOW);
  noTone(BUZZER_PIN);
}

void LEDGreen() {
  // MENGHIDUPKAN LAMPU HIJAU DAN BUZZER
  digitalWrite(LED_PIN3, HIGH);
  tone(BUZZER_PIN, 1500);

  // DELAY 1 DETIK
  delay(300);

  // MEMATIKAN LAMPU HIJAU DAN BUZZER
  digitalWrite(LED_PIN3, LOW);
  noTone(BUZZER_PIN);
}

```

Terakhir kita harus menghubungkan Arduino dengan sistem wms agar pendeteksian barang dapat berjalan, itu dikarenakan data barang yang dibuat berada pada database yang berada pada sistem. Cara menghubungkan dengan sistem yaitu menggunakan kabel lan untuk dapat mentransfer data yang akan dideteksi.

Setelah memasukkan *source code* tersebut kemudian melakukan *verify* untuk memeriksa ada yang error atau tidak, jika tidak kemudian klik *upload* untuk menginputkan ke mikrokontroler arduino. Setelah di *upload* maka alat akan bekerja dengan sendirinya dan sesuai dengan perintah yang telah di instruksikan, dan jangan lupa untuk menyesuaikan port Arduino terlebih dahulu.

Selanjutnya proses pengcodean pada alat conveyor. Pertama kita melakukan pendefinisian pin

```

int trig= 12; // membuat variabel trig yang di set ke-pin A0
int echo= 11; // membuat variabel echo yang di set ke-pin A1

int en1 = 8 ;
int in3 = 10;
int in4 = 9 ;

int echol= 2;
int trigl= 3;

int en0 = 4;
int in5 = 6;
int in6 = 5;

```

Selanjutnya kita melakukan inisialisasi dan memberikan nilai awal pada void setup

```

/**
  Initialize.
 */
void setup() {
  Serial.begin (9600);
  pinMode(trig, OUTPUT);
  pinMode(echo, INPUT);
  pinMode(in3, OUTPUT);
  pinMode(in4, OUTPUT);
  pinMode(en1, OUTPUT);

  Serial.begin (9600);
  pinMode(trig1, OUTPUT);
  pinMode(echo1, INPUT);
  pinMode(in5, OUTPUT);
  pinMode(in6, OUTPUT);
  pinMode(en0, OUTPUT);

}

```

Setelah kita melakukan penyetingan dengan menginisialisasi dan memberikan nilai awal pada void setup, selanjutnya kita dapat melanjutkan tahapan pada void loop.

```

/**
  Main loop.
*/
void loop () {
  float duration, distance;
  digitalWrite(trig, LOW);
  delay(1000);
  digitalWrite(trig, HIGH);
  delayMicroseconds(10);
  digitalWrite(trig, LOW);
  duration = pulseIn(echo, HIGH);
  distance = (duration/2) * 0.03448;

  Serial.print("Jarak benda 1 = ");
  Serial.print(distance);
  Serial.println(" cm");

  if (distance > 0 && distance <= 5){ //mengatur jarak pendeteksian barang
    digitalWrite(in3, LOW);
    digitalWrite(in4, HIGH);
    analogWrite(en1, 150);
    delay(10); // waktu pada saat mendeteksi barang
  }else{
    digitalWrite(in3, LOW);
    digitalWrite(in4, LOW);
    analogWrite(en1, 150);
  }
  delay(500);
}

```

```

digitalWrite(trig1, LOW);
delay(1000);
digitalWrite(trig1, HIGH);
delayMicroseconds(10);
digitalWrite(trig1, LOW);
duration = pulseIn(echol, HIGH);
distance = (duration/2) * 0.03448;

Serial.print("Jarak benda 2 = ");
Serial.print(distance);
Serial.println(" cm");

if (distance > 0 && distance <= 5){ //mengatur jarak pendeteksian barang
    digitalWrite(in5, LOW);
    digitalWrite(in6, HIGH);
    analogWrite(en0, 150);
    delay(10); // waktu pada saat mendeteksi barang
}else{
    digitalWrite(in5, LOW);
    digitalWrite(in6, LOW);
    analogWrite(en0, 150);
}
delay(500);
}

```

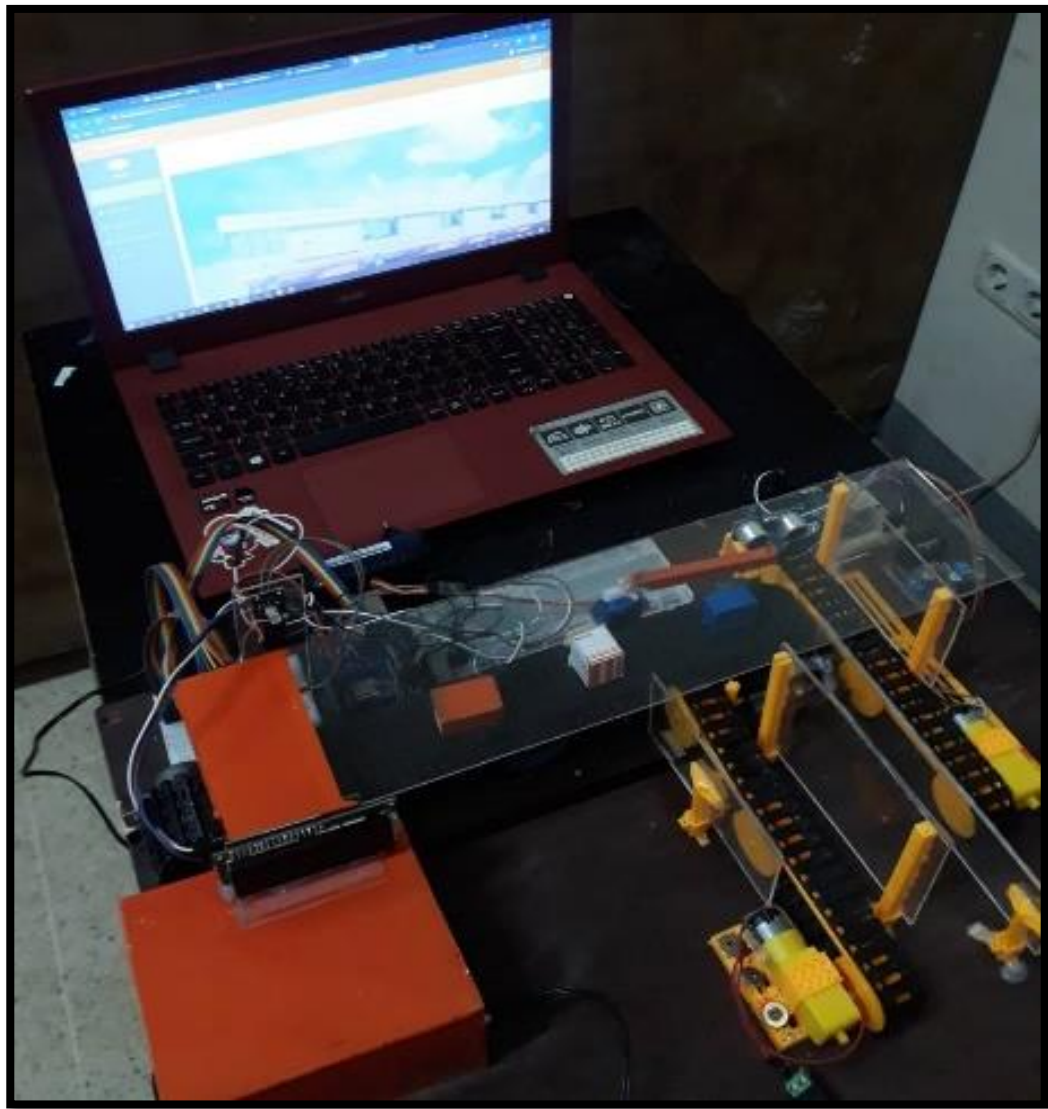
Disini melakukan 2 kali pengulangan code karena menggunakan 2 sensor dan 2 conveyor.

Jika telah selesai melakukan pengkodean maka langkah selanjutnya kita harus mengverifikasi code untuk memeriksa error, jika tidak ada error maka selanjutnya kita langsung meng-upload ke microcontrler Arduino.

## Implementasi Alat

Hasil dari penelitian ini dapat dilihat pada gambar.









## Daftar Pustaka

- [1] Streufert dan B. P, Automatically adjustable slide retract for conveyor, US Patent 9,248,983: Google Patents, 2016.
- [2] Pane, Syafrial Fachri and Awangga, R. M. a. Azhari dan B. R. , “Qualitative Evaluation of RFID Implementation on Warehouse Management System,” *Telkomnika*, vol. 16, 2018.
- [3] Pane, Syafrial Fachri and Awangga, Rolly Maulana and Azhari, Bayu Rahmad and Tartila dan G. Romadhanu, “RFID-based conveyor belt for improve warehouse operations,” *Telkomnika*, vol. 2, pp. 794--800, 17.

