

CHAPTER 5

IMPLEMENTATION AND TESTING

5.1 Implementation

In this project, the application used to create prototype sort of delivery package is Arduino IDE application. And the devices used in this prototype include: Arduino Mega, ultrasonic sensor, laser sensor, ldr sensor, load cell sensor, LCD display, DC geared motor, and a conveyor. Here are some program snippets / snippets of components and sensors used in this prototype project.

```
1  #include "HCSR04.h"
2  #include <HX711_ADC.h>
3  #include <LiquidCrystal.h>
4
5  const int rs = 14, en = 15, d4 = 16, d5 = 17, d6 = 18, d7 = 19;
6  LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
7
8  #define KonversiVolumetrik 5000
9
10 #define ledReady 12
11
12 #define MOTOR_E 2
13 #define MOTOR_IN1 3
14 #define MOTOR_IN2 4
15
16 #define laser 5
17
18 #define trigerKanan 6
19 #define echoKanan 7
20 #define trigerKiri 8
21 #define echoKiri 9
22
23 #define trigerAtas 11
24 #define echoAtas 10
25
26 HCSR04 sensorKiri(trigerKiri, echoKiri);
27 HCSR04 sensorKanan(trigerKanan, echoKanan);
28
29 HCSR04 sensorAtas(trigerAtas, echoAtas);
30
31 HX711_ADC LoadCell(A1, A0);
32
33 void jalanmotor(int speed, boolean arah);
34
35 float ukurLebar();
36 float ukurTinggi();
37
38 unsigned long waktuMulai;
```

Illustration 5.1: Declaration and Initialization

(Illustration 5.1)

Code for pin declaration used as input pin and output pin and also function declaration. Among other things: ultrasonic sensor, laser sensor and ldr sensor, load cell sensor, DC geared motor, led, and also LCD display, wide measurement function, high measuring function, and function run motor conveyor.

```
218 void jalanmotor(int speed, boolean arah)
219 {
220     analogWrite(MOTOR_E, speed);
221     if (arah == true)
222     {
223         digitalWrite(MOTOR_IN1, HIGH);
224         digitalWrite(MOTOR_IN2, LOW);
225     }
226     else
227     {
228         digitalWrite(MOTOR_IN1, LOW);
229         digitalWrite(MOTOR_IN2, HIGH);
230     }
231 }
232 
```

Illustration 5.2: Function jalanmotor()

(Illustration 5.2)

Code function to set the direction of rotation of a DC geared motor used as a conveyor drive.

```
104 unsigned long ms = millis();
105
106 while (digitalRead(laser) == HIGH)
107 {
108     jalanmotor(70, false);
109 }
110
111 ms = millis() - ms;
112
113 float p, l, t, v;
114 float beratVolumetrik;
115
116 p = (float)ms*0.0226293103448276;
```

Illustration 5.3: Calculation of ms Value (Packet Length)

(Illustration 5.3)

Code program to calculate packet length using laser sensor and ldr sensor.



```

237 float ukurLebar()
238 {
239     float offsetKanan = -12.0f;
240     float offsetKiri = 0.0f;
241     float offsetAkhir = 4.0f;
242
243     float jarakKiri = sensorKiri.read(HCSR04::MetricsEL::mm) + offsetKiri;
244     float jarakKanan = sensorKanan.read(HCSR04::MetricsEL::mm) + offsetKanan;
245
246     if (jarakKiri > 0 && jarakKanan > 0)
247     {
248         float lebar = (248.0f - jarakKanan - jarakKiri - 40.0f);
249
250         if (lebar < 0)
251         {
252             return 0;
253         }
254
255         return (lebar + offsetAkhir) / 10;
256     }
257
258     else
259     {
260         return 0;
261     }
262 }
263
264 }
265
266

```

Illustration 5.4: Function ukurLebar()

(Illustration 5.4)

The program code to calculate the width of the packet using an ultrasonic sensor.

```

268 float ukurTinggi()
269 {
270
271     float tinggiBenda = sensorAtas.read(HCSR04::MetricsEL::mm);
272     float tinggi = (120.0f - tinggiBenda) / 10;
273
274     if (tinggi < 0)
275     {
276         tinggi = 0;
277     }
278
279     return tinggi;
280 }
281

```

Illustration 5.5: Function ukurTinggi()

(Illustration 5.5)

Code program for calculating packet height using ultrasonic sensor.

```

152 float m;
153
154 for (int i = 0; i < 1050; i++)
155 {
156     LoadCell.update();
157     m = LoadCell.getData();
158
159     delay(1);
160 }
161
162 m=m/1000.0f;
163
164 Serial.print("Massa: ");
165 Serial.print(m);
166 Serial.println(" kg");

```

Illustration 5.6: Calculation of Actual Weight (Load Cell)

(Illustration 5.6)

The code to process the mass or weight calculation using a load cell sensor that has been designed into a weigh board.

```

172     lcd.clear();
173
174     lcd.setCursor(1,0);
175
176     lcd.print((int)p);
177
178     lcd.print(";");
179
180     lcd.print((int)l);
181
182     lcd.print(";");
183
184     lcd.print((int)t);
185
186     lcd.print("->");
187
188     lcd.print((float)beratVolumetrik,3);
189     lcd.print("kg");
190
191     lcd.setCursor(1,1);
192
193     lcd.print((float)m,5);
194     lcd.print(" kg");

```

Illustration 5.7: Display the Counting Result to 16x2 LCD Display Screen

(Illustration 5.7)

Code to display the results of packet volume measurements, packet volumetric weight conversion, and packet mass or weight.

```

122     float beratVolumetrik;
123     beratVolumetrik = (p*l*t)/KonversiVolumetrik;

```

Illustration 5.8: Volumetric Conversion Formula

(Illustration 5.8)

This is the variable declaration for the conversion process from volume (cm^3) to weight (gram).

```

198         if (beratVolumetrik > m)
199         {
200             lcd.setCursor(0,0);
201         }
202
203         else
204         {
205             lcd.setCursor(0,1);
206         }
207
208         lcd.write(0b01111110);

```

Illustration 5.9: Decision Determination

(Illustration 5.9)

To determine which provisions to choose, actual weight or volumetric weight.

```

113         float p, l, t, v;
114
115
116         p = (float)ms*0.0226293103448276;
117         l = ukurLebar();
118         t = ukurTinggi();
119
120         v = (l * t * p);

```

Illustration 5.10: Variable Declaration P, L, T, V & Call Function ()

(Illustration 5.10)

The process of declaration of variable length (p), width (l), height (t), and also volume (v).

5.2 Testing

Here is the testing process prototype package sorting system. In this test process, objects used as packets are objects in the form of boxes / cubes / cubes that have a flat surface. It also has mass or weight. Because, before the test process begins, the object must be loaded first so that the object has a weight / mass so that later it can be calculated by weight on the process of calculating the mass of the object (weighing board). Because if the package has no actual weight (no real mass / weight), then at the end of the process it will be difficult to compare with the volumetric weight.



Illustration 5.11: Laying Packets

Prepared items / packets are then placed at the end of the conveyor when the led light is on (alert for the officer to immediately place the package) and the conveyor is still dead for a while until the conveyor is on.



Illustration 5.12: Laser Light Detects Packets (Front)

Once the package is placed on top of the conveyor and conveyor then turns on / runs, the front end of the packet will begin to touch the laser beam. And from there, the process of calculating the long value begins.



Illustration 5.13: Laser Light Detects Packets

The packet is still running when the laser beam touches part of part of the package, to find the value of ms (milisecond) from the beginning of the laser until the laser does not detect the existence of objects / packets that run through the laser beam.



Illustration 5.14: Laser Light Detects Packets (Rear)

The back end of the packet is exposed to the laser, it means that the search and counting of ms (milisecond) values has been done successfully. The initial ms value is the value of ms obtained when the packet first touches the laser beam. And the final ms value is the value of ms obtained when the laser beam touches the back end of the package. Thus, the value of ms to be used as a long value, is obtained by calculating the ms value from the front end of the package to the end of the back of the packet.

And as long as the packet touches the laser until it is not exposed to the laser again, the calculation process within the project not only measures the ms (milli seconds) value, but also calculates the value of packet width and value using ultrasonic sensors located on the right, left and upper side of the tunnel.

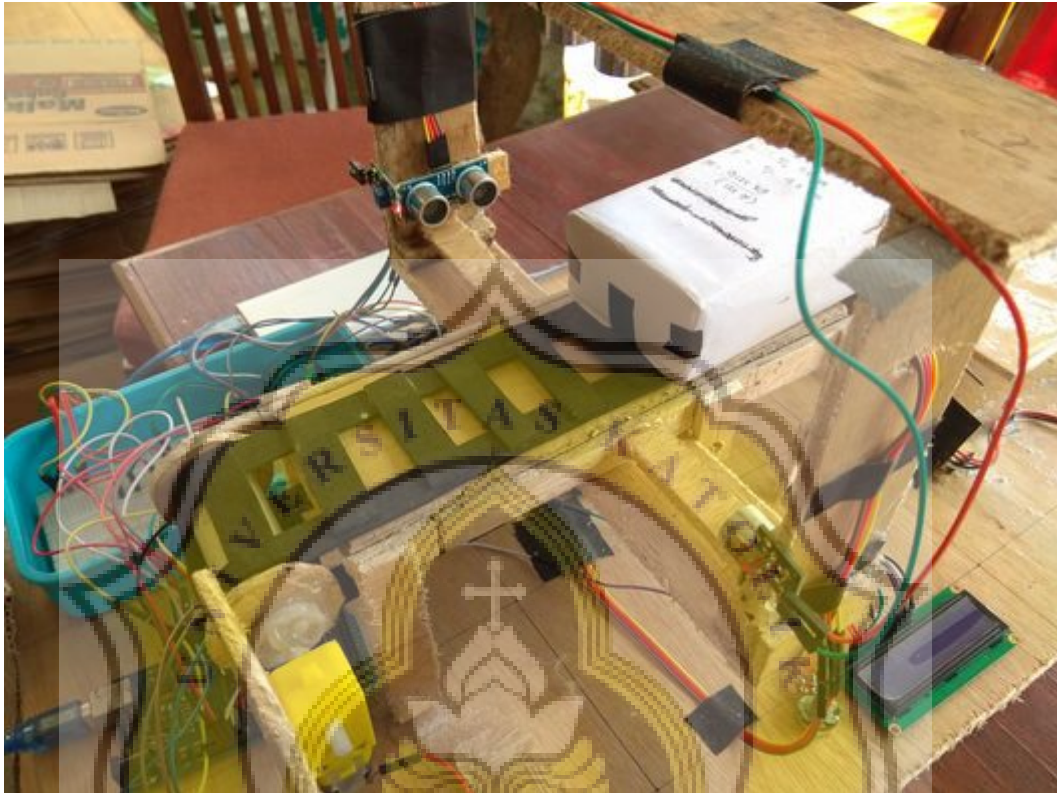


Illustration 5.15: Volume Count is Complete

After the volume counting process is complete, then the packet will proceed to the weigh board.

After calculating the volume value is done, then the packet will be forwarded to weighing board to calculate the actual weight. The package above the conveyor will fall to the weighing board and after that the actual weight calculation process will begin.



Illustration 5.16: Packet Dropped to Weigh Board

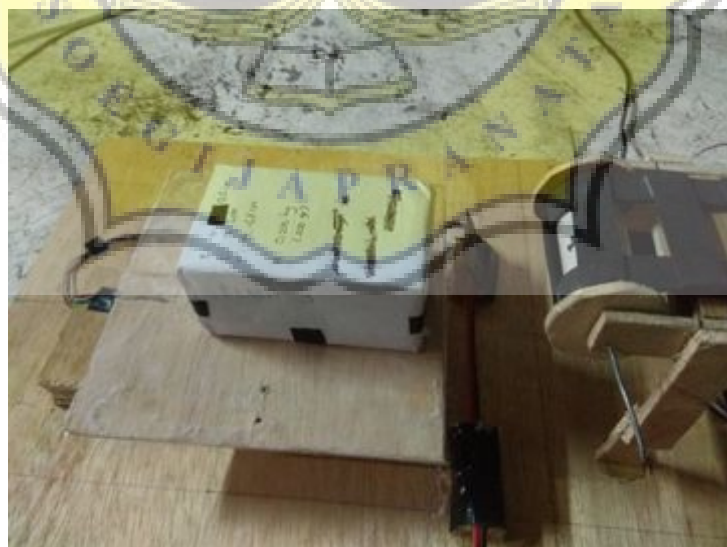


Illustration 5.17: Calculate Actual Weight



Illustration 5.18: Calculation Result in Serial Monitor

The result of calculating the value of P, L, T, Volume, Volumetric Weight and Mass Weight / Actual shown in Serial Monitor.

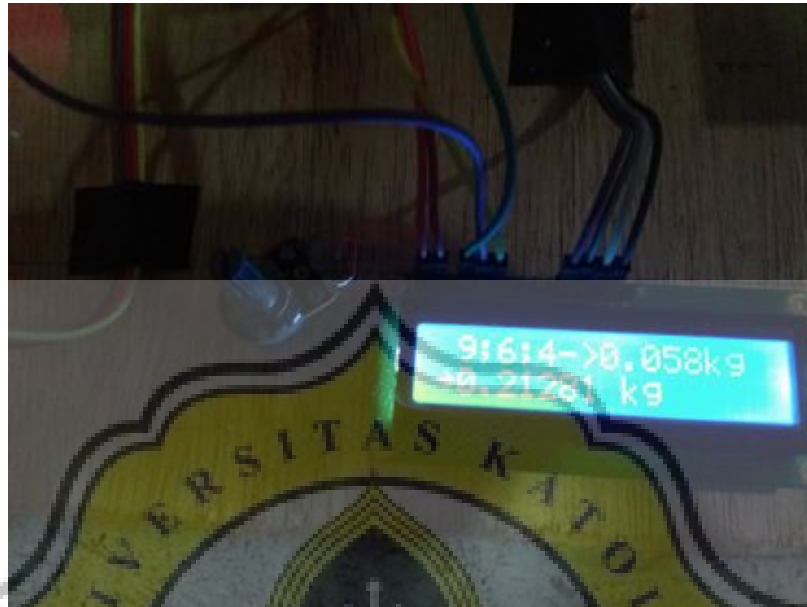


Illustration 5.19: The Results of All Calculations

The results of the calculation of the value of length, width value, high value, conversion of volumetric weight value, actual weight value and arrow arrow as a decision taken.