CHAPTER 5
IMPLEMENTATION AND RESULTS

5.1. Implementation

Arduino IDE is made from the JAVA programming language. Arduino IDE is also equipped with a C/C++ library commonly called Wiring which makes input and output operations easier. This Arduino IDE was developed from Processing software which was remodeled into a special Arduino IDE for programming with Arduino.

```
1.  #include <Wire.h>
2.  #include <LiquidCrystal_I2C.h>
3.  #include <Servo.h>
4.  #include <Fuzzy.h>
5.  #include "HX711.h"
```

We will first set up the libraries we need first. Wire.h is used so that we can use the features of i2c easily. LiquidCrystal_I2C.h is a library for LCD use. Servo.h is a library for using servos. Fuzzy.h is the algorithm used in this project, namely fuzzy mamdani. Hx711.h is a library for load sensor usage.

```
6.  #define PIR   6
7.  #define Echo  5
8.  #define Trig  4
9.  #define LOADCELL_DOUT_PIN  A0
10. #define LOADCELL_SCK_PIN   A1
11.
12. LiquidCrystal_I2C lcd(0x27, 16, 2);     //inisialisasi lcd (alamat i2c, ukuran lcd)
13. Servo Tutup;                            //inisialisasi servo dgn nama tutup
14. HX711 Scale;                            //inisialisasi sensor berat dgn nama scale
```

The code in line 12 serves to initialize the lcd, 0x27 is the i2c address and 16.2 is the lcd size (16 columns 2 rows). On lines 13 and 14 are the servo initialization and the weight sensor.

```
15.
16.     //*************** FUZZY ***************//
17.     Fuzzy *fuzzy = new Fuzzy();
18.
```

```
19.  //Fuzzy Jarak
20.  FuzzySet *Tinggi = new FuzzySet(30, 30, 60, 120);
21.  FuzzySet *Cukup = new FuzzySet(60, 120, 120, 180);
22.  FuzzySet *Rendah = new FuzzySet(120, 180, 210, 210);
23.
24.  //Fuzzy Beban
25.  FuzzySet *Ringan = new FuzzySet(0, 0, 500, 1500);
26.  FuzzySet *Sedang = new FuzzySet(500, 1500, 1500, 2500);
27.  FuzzySet *Berat = new FuzzySet(1500, 2500, 3000, 3000);
28.
29.  //Fuzzy Output
30.  FuzzySet *Sedikit = new FuzzySet(0, 0, 20, 40);
31.  FuzzySet *Lumayan = new FuzzySet(20, 40, 60, 80);
32.  FuzzySet *Penuh = new FuzzySet(60, 80, 100, 100);
```

In lines 17 to 32, we will set the curves we need in the fuzzy calculation. For the Altitude and weight curves, we will use the triangular membership function. For the output or capacity curve, we will use a trapezoidal membership function.

```
33.  float deffuzy = 0, y1 = 810, y2 = 1883, y3 = 3198;
34.  uint8_t a1 = 40, a2 = 40, a3 = 40;
35.  const String Satuan = " %  ";                  //variabel satuan persen
36.  String Kondisi = "";                  //variael kondisi dengan nilai awal ""
37.  uint32_t Durasi = 0; //            //durasi hitung jarak dari sensor
38.  uint8_t JarakCm, dJarakCm;                  //variabel nilai jarak
39.  float Persen;                  //variabel persen
40.  boolean sensorValue;                  //variabel penyimpan nilai sensor PIR
41.
42.  unsigned long previousMillis = 0;
43.  const long interval = 1000;
44.  float berat;
```

Variables in some parts serve to store the input value of the sensors used. For line 33 which is used to help equalize the defuzzification process of the manual count with the Arduino count.

```
45. void setup() {
46.  Serial.begin(9600);
```

Line 46 is for start serial monitor

```
47.   //Inisialisasi Pin
48.   pinMode(PIR, INPUT);
49.   pinMode(Trig, OUTPUT);
50.   pinMode(Echo, INPUT);
51.   Tutup.attach(9);                    //Pin 9 sebagai output servo
52.   Tutup.write(0);                     //Kondisi servo sudut 0
53.
54.   Scale.begin(LOADCELL_DOUT_PIN, LOADCELL_SCK_PIN);
55.   Scale.set_scale(480.f);             //nilai hasil dr kalibrasi
56.   Scale.tare();                       // reset the scale to 0
57.   lcd.setCursor(0, 1);                //set kolom, baris 0,1
58.
59. fuzzy->setInput(1, dJarakCm);         //input nilai Jarak ke kurva 1 (jarak)
60.    fuzzy->setInput(2, load);          //input nilai Beban ke kurva 2 (beban)
61.    fuzzy->fuzzify();                  //fuzzyfikasi
```

Line 59 works for inputting Distance values in fuzzy calculations. Line 60 works for inputting weight values in fuzzy calculations. Line 61 will start in the process of fuzzification.

```
62. deffuzy = ((y1 * Sedikit->getPertinence()) + (y2 * Lumayan->getPertinence()) +(y3 * Penuh->getPertinence())) / ((a1 * Sedikit->getPertinence()) + (a2 * Lumayan->getPertinence()) + (a3 * Penuh->getPertinence()));
63.    float output = deffuzy;
64.    Persen = map(output, 20.25, 79.95, 0, 100.00);
65. if (Sedikit->getPertinence() > Lumayan->getPertinence() && Penuh->getPertinence() == 0) {
```

```
66.        Kondisi = "Sedikit";
67.      }
```

Lines 62 is the process of calculating defuzzification. For each crrisp calculation that requires a random value, the variables y1, y2 and y3 are used so that each crisp calculation will use the same total random value. From the curve made at the beginning, it has been tested that the maximum fuzzy value will be 79.95. The minimum value will be 20.25. So line 91 serves to match the fuzzy calculation value with the output on the LCD. Lines 65 to 67, serve to display the results of the capacity variable and have been adjusted to the existing rulebase.

```
68.      //1
69.     FuzzyRuleAntecedent *RendahRingan = new FuzzyRuleAntecedent();
70.     RendahRingan->joinWithAND(Rendah, Ringan);
71.
72.     FuzzyRuleConsequent *Hasil1 = new FuzzyRuleConsequent();
73.     Hasil1 ->addOutput(Sedikit);
74.
75.     FuzzyRule *fuzzyRule1 = new FuzzyRule(1, RendahRingan, Hasil1);
76.     fuzzy->addFuzzyRule(fuzzyRule1);
```

The rule base on lines 68 to 76 is an example of the rule base low height and light weight equal to little.

5.2. Testing

Testing the weight sensor here uses 2 measuring tools. The weight sensor itself and the regular scale as reference values. This comparison was conducted to determine the level of measurement accuracy of the TAL220B weight sensor with the hx711 module.

Object	HX711	Scales	Error%
Mineral water	491 g	500 g	1.8
Blackmores	170 g	175 g	2.8
Nivea	263 g	260 g	1.1
3 pcs Smartphones	639 g	650 g	1.6
Kanebo	207 g	200g	3.5
Parfume	331 g	320 g	3.4
Smartphones Box	199 g	210 g	5.2
Mineral Water + 3 pcs Smartphones	1124 g	1150 g	2.2
Mineral water + parfume	806 g	820 g	1.7
Mineral Water + 3 pcs Smartphones + Parfume	1415 g	1470 g	3.7
Mineral Water + 3 pcs Smartphones + Parfume +Nivea	1702 g	1730g	1.6
Mineral Water + Blackmores + Nivea	948 g	935 g	1.3
3 pcs Smartphones + Blckmores + Nivea	1042 g	1075 g	3.0
Mineral Water + 3 pcs Smartphones + Parfume + Nivea + Kanebo	1903 g	1930 g	1.3
Mineral Water + 3 pcs	2087 g	2140 g	2.4

Smartphones + Parfume + Nivea + Kanebo +Smartphones box			
	Total Error%		36.6
	Average Error%		2.4%

Table 5.1 : Table result of load cell testing

Load Cell Accuracy = 100% - 2.4% = 97.6 %

The calculation used to find the error value, namely the interval from the load sensor value and scales will be divided by the value of the scales and the last step is multiplied by 100%. From the test results and the calculation of the %error value against the load sensor reading value, it can be concluded that the sensor can work well because the %error value for 15 measurements is very small, which is only 2.4 %.

Testing the weight sensor here uses 2 measuring tools. The HC-SR04 itself and ruler as reference values. This comparison was conducted to determine the level of measurement accuracy of the HC-SR04 Ultrasonic distance measuring transducer sensor.

Object	HC-SR04	Ruler	Error %
Mineral water	200mm	200mm	0
Blackmores	93mm	95mm	2.1
Nivea	185mm	185mm	0
3 pcs Smartphones	40mm	42mm	4.7
Kanebo	169mm	170mm	0.5
Parfume	122mm	125mm	2.4
Smartphones box	177mm	180mm	1.6
Mouse	37mm	40mm	7.5
Tissue basah	32mm	30mm	6.6
Tissue	79mm	80mm	1.2

Aloe vera	51mm	50mm	2
Blue band	53mm	55mm	3.6
Betadine	142mm	145mm	2.0
Garnier	77mm	76mm	1.3
Nutela	92mm	90mm	2.2
	Total Error%		37.7
	Average Error%		2.5

Table 5.2 : Table result of HC-SR04 testing

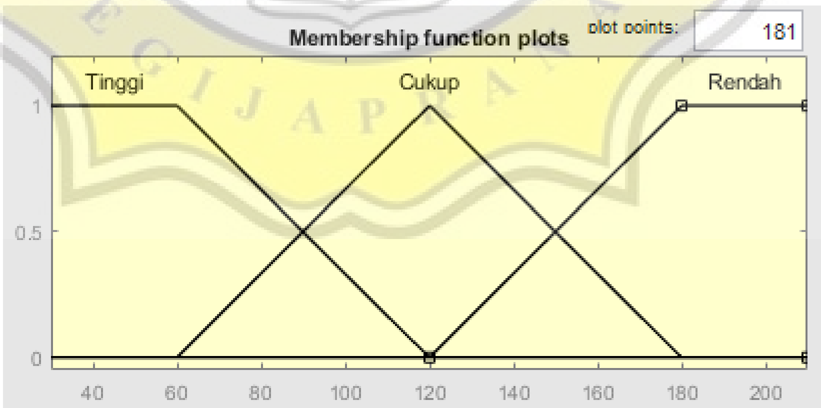
HC-SR04 Accuracy = 100% - 2.5% = 97.5 %

The calculation used to find the error value, namely the interval from the HC-SR04value and ruler value will be divided by the value of the ruler and the last step is multiplied by 100%. From the test results and the calculation of the %error value against the HC-SR04 reading value, it can be concluded that the sensor can work well because the %error value for 15 measurements is very small, which is only 2.35%.

Testing Method Using Integral

Sample : Object weighing 1200grams and height 80mm

1. Fuzzyfikasi



Picture Kurva Variabel Ketinggian

The object height variable must be subtraction with 210mm first, because the data obtained by the ultrasonic sensor is the distance from the sensor to the object. So when the object's height is 125mm, then the x value will be obtained from subtracting 210mm by 125mm.

$$\mu\ \text{Rendah} = \frac{x - 120}{180 - 120}$$

$$\mu\ \text{Rendah} = \frac{130 - 120}{180 - 120}$$

$$\mu\ \text{Rendah} = \frac{10}{60}$$

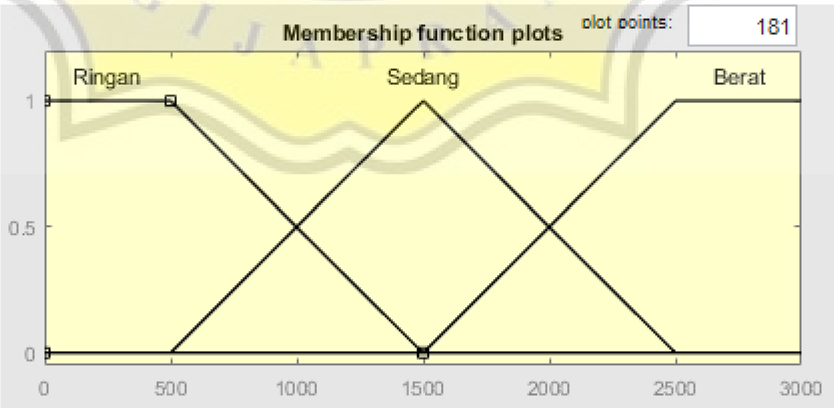
$$\mu\ \text{Rendah} = 0.16$$

$$\mu\ \text{Cukup} = \frac{180 - x}{180 - 120}$$

$$\mu\ \text{Cukup} = \frac{180 - 130}{180 - 120}$$

$$\mu\ \text{Cukup} = \frac{50}{60}$$

$$\mu\ \text{Cukup} = 0.83$$



Picture Kurva Variabel Berat

$$\mu \text{ Ringan} = \frac{1500 - x}{1500 - 500}$$

$$\mu \text{ Rringan} = \frac{1500 - 1200}{1500 - 500}$$

$$\mu \text{ Ringan} = \frac{300}{1000}$$

$$\mu \text{ Ringan} = 0.3$$

$$\mu \text{ Sedang} = \frac{x - 500}{1500 - 500}$$

$$\mu \text{ Sedang} = \frac{1200 - 500}{1500 - 500}$$

$$\mu \text{ Sedang} = \frac{700}{1000}$$

$$\mu \text{ Sedang} = 0.7$$

2. Inferensi

Ketinggian / Berat	Tinggi	Cukup	Rendah
Ringan	Penuh	Sedikit	Sedikit
Sedang	Penuh	Lumayan	Sedikit
Berat	Penuh	Penuh	Lumayan

*Conjuction :

IF Ketinggian (Cukup) = 0.83 AND Berat (Sedang) = 0.7 THEN Kapasitas (Lumayan) = 0.7

IF Ketinggian (Rendah) = 0,16 AND Berat (Sedang) = 0.7 THEN Kapasitas (Sedikit) = 0.16

IF Ketinggian (Cukup) = 0.83 AND Berat (Ringan) = 0.3 THEN Kapasitas (Sedikit) = 0.3

IF Ketinggian (Rendah) = 0.16 AND Berat (Ringan) = 0.3 THEN Kapasitas (Sedikit) = 0.16

*Disjunction:

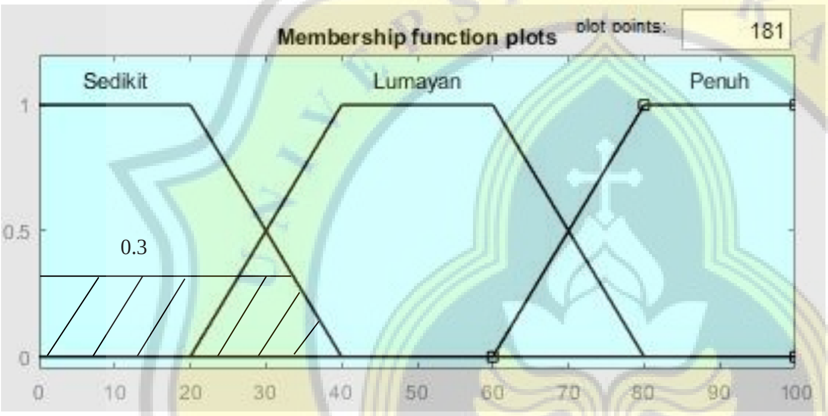
Capacity : 1. Lumayan 0.7

2. Sedikit 0.3

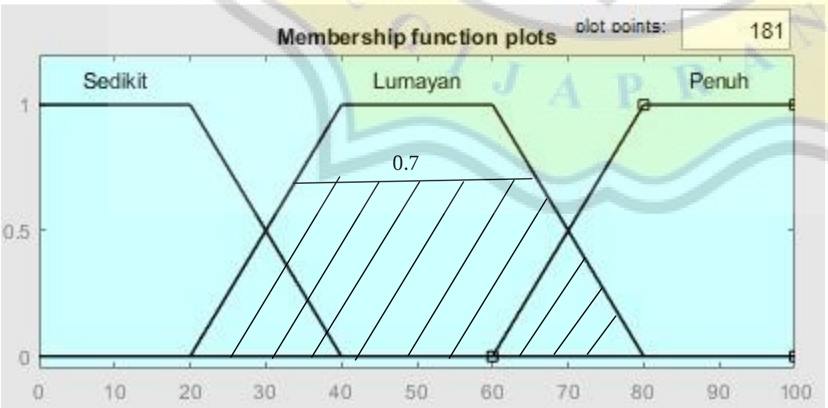
3. Deffuzifikasi

a. Clipping

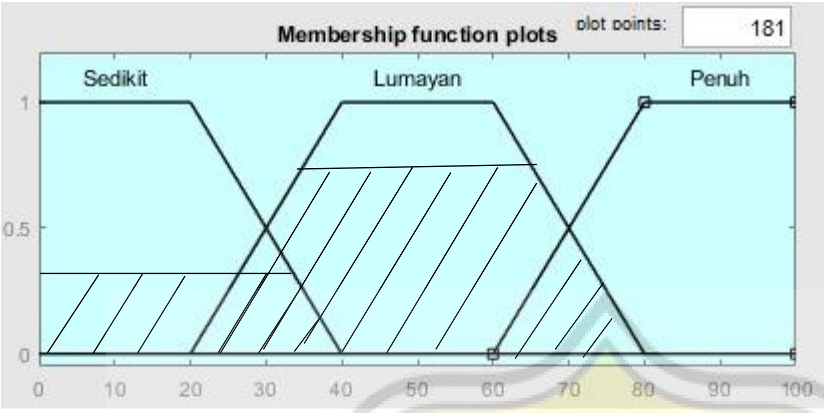
Capacity Sedikit 0.3



Capacity Lumayan 0.7



b.Crisp



$$x^* = \frac{\int y\mu_R(x)dq}{\int \mu_R(x)dq} => \frac{Momen}{Luas}$$

$$\frac{t1-20}{40-20} = 0.3 => t1 = 20 + (0.3 * 20) = 26$$

$$\frac{t2-20}{40-20} = 0.7 => t2 = 20 + (0.7 * 20) = 34$$

$$\frac{80-t3}{80-60} = 0.7 => t3 = 80 - (0.7 * 20) = 66$$

$$t4 = 80$$

$M1 = \int_0^{26} 0.3x\, dx = 101{,}4$ $\quad A1 = \int_0^{26} 0.3\, dx = 7{,}8$

$M2 = \int_{26}^{34} \frac{x-26}{34-26} x\, dx = 125{,}3$ $\quad A2 = \int_{26}^{34} \frac{x-26}{34-26}\, dx = 4$

$M3 = \int_{34}^{66} 0.7\, x\, dx = 1120$ $\quad A3 = \int_{34}^{66} 0.7\, dx = 22{,}4$

$M4 = \int_{66}^{80} \frac{80-x}{80-66} x\, dx = 494{,}6$ $\quad A4 = \int_{66}^{80} \frac{80-x}{80-66}\, dx = 7$

$$x^* = \frac{M1 + M2 + M3 + M4}{A1 + A2 + A3 + A4}$$

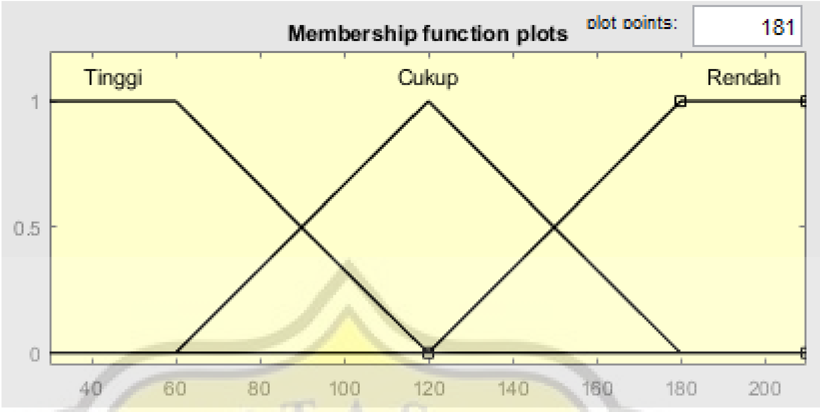
$$x^* = \frac{101{,}4 + 125{,}3 + 1120 + 494{,}6}{7{,}8 + 4 + 22{,}4 + 7}$$

$$x^* = 44.03 \text{ (lumayan)}$$

Testing Method Without Integral

Sample : Object weighing 320 grams and height 135mm

1. Fuzzyfikasi



Picture Kurva Variabel Ketinggian

$$\mu\ \text{Cukup} = \frac{x - 60}{120 - 60};$$

$$\mu\ \text{Cukup} = \frac{75 - 60}{120 - 60}$$

$$\mu\ \text{Cukup} = \frac{15}{60}$$

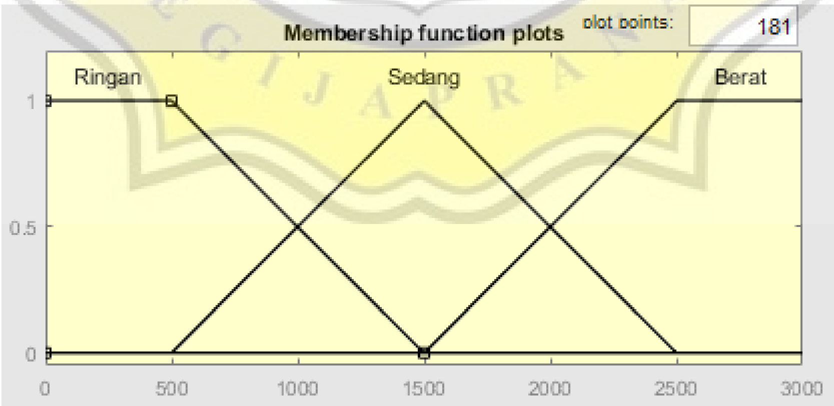
$$\mu\ \text{Cukup} = 0.25$$

$$\mu\ \text{Tinggi} = \frac{120 - x}{120 - 60}$$

$$\mu\ \text{Tinggi} = \frac{120 - 75}{120 - 60}$$

$$\mu\ \text{Tinggi} = \frac{45}{60}$$

$$\mu\ \text{Tinggi} = 0.75$$



Picture Kurva Variabel Berat

$$\mu \text{ Ringan } [x] = \begin{cases} 1; & x \leq 500 \\ \frac{1500 - x}{1500 - 500}; & 500 \leq x \leq 1500 \\ 0; & x \geq 1500 \end{cases}$$

$$\mu \text{ Ringan} = 320 \leq 500$$

$$\mu \text{ Ringan} = 1$$

2. Inferensi

Ketinggian / Berat	Tinggi	Cukup	Rendah
Ringan	Penuh	Sedikit	Sedikit
Sedang	Penuh	Lumayan	Sedikit
Berat	Penuh	Penuh	Lumayan

*Conjuction :

IF Ketinggian (Cukup) = 0.25 AND Berat (Ringan) = 1 THEN Kapasitas (Sedikit) = 0.25

IF Ketinggian (Tinggi) = 0,75 AND Berat (Ringan) = 1 THEN Kapasitas (Penuh) = 0.75

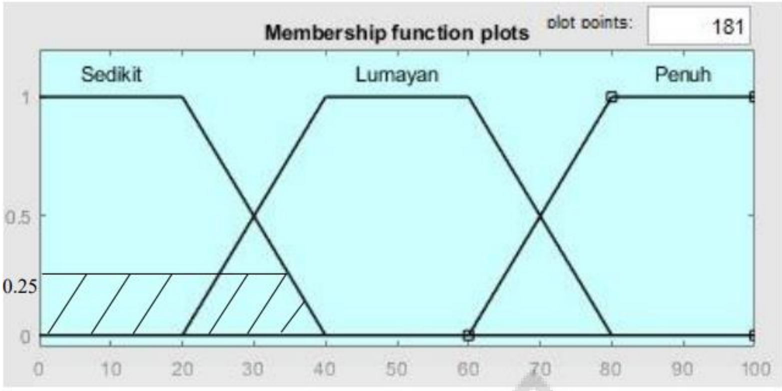
*Disjunction:

Capacity : 1. Sedikit 0.25

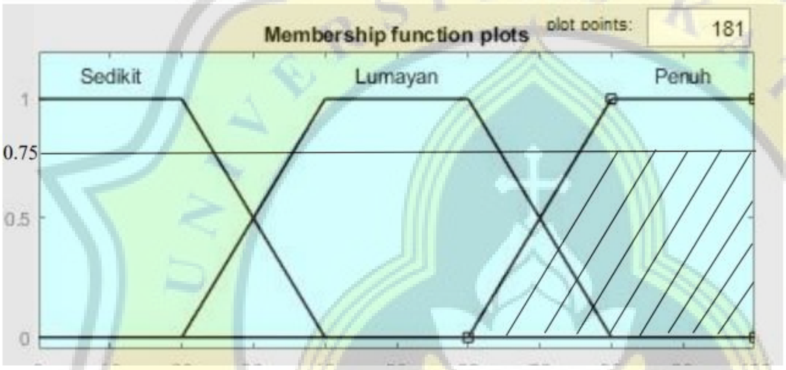
2. Penuh 0.75

a. Clipping

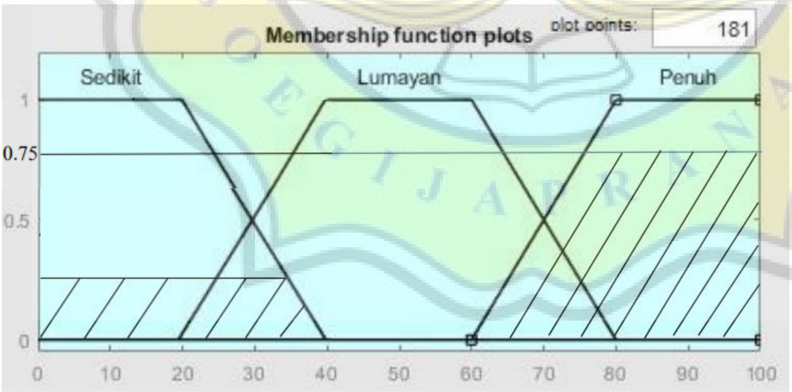
Capacity Sedikit 0.25



Capacity Penuh 0.75



b.Crisp



$$Y^* = \frac{810 \ x \ 0.25 + \ 3198 \ x \ 0.75}{40 \ x \ 0.25 + 40 \ x \ 0.75}$$

$$Y^* = \frac{2601}{40}$$

Y* = 65.02 (Penuh)

Next is testing the fuzzy calculating program, this test aims to find out whether the system with the application of this fuzzy method can produce output in accordance with the design that has been made previously. The test results can be seen in the table.

Tinggi (asli)	Berat (asli)	Defuzzyfikasi (Arduino)	Defuzzyfikasi (Manual)	Tingkat Kepenuhan	Error %
200mm	500g	79.95	79.95	Penuh	0
125mm	320g	55.08	54.27	Penuh	1.4
95mm	175g	25.22	25.02	Sedikit	0.7
170mm	200g	79.95	79.95	Penuh	0
90mm	150g	20.25	20.25	Sedikit	0
80mm	1200g	39.03	39.02	Lumayan	0.02
76mm	1500g	40.82	40.90	Lumayan	0.1
55mm	720g	27.60	27.62	Sedikit	0.07
135mm	320g	65.02	65.02	Penuh	0
0(kosong)	0(kosong)	20.25	20.25	Sedikit	0
			Total Error		2.29
			Average Error		0.22

Table 5.3 : Table result of fuzzy calculation with real height and weight

For the results of fuzzy testing with manual object measurements, good calculation accuracy is obtained. The interval between manual calculations and calculations from Arduino gets an average error of 0.22 %. This results show that the fuzzy inference system that has been run properly and produces output can meet the rules that have been designed.

Tinggi (Sensor)	Berat (Sensor)	Defuzzyfikasi (Arduino)	Defuzzyfikasi (Manual)	Tingkat Kepenuhan	Error %
188mm	485g	79.95	79.95	Penuh	0
124mm	205g	54.08	54.27	Penuh	0.3
90mm	135g	20.25	20.25	Sedikit	0
174mm	155g	79.95	79.95	Penuh	0
91mm	123g	21.24	21.44	Sedikit	0.9
78mm	1180g	38.49	38.49	Lumayan	0
75mm	1425g	40.37	40.36	Lumayan	0.02
53mm	700g	26.82	26.79	Sedikit	0.1
134mm	260g	64.03	63.83	Penuh	0.3
6mm(kosong)	0(kosong)	20.25	20.25	Sedikit	0
			Total Error		1.62
			Average Error		0.16

Table 5.4 : Table result of fuzzy calculation sensor measurement

Based on the test in the table above and the analysis of fuzzy inference system calculations in real conditions through ultrasonic readings and weight sensors, the results show that the fuzzy inference system that has been run properly and produces output can meet the rules that have been designed. The table shows that an empty trash can when compared to trash weighing 135 grams and a height of 90 mm can produce the same output. The 2 tables above also show that from manual measurements and sensor measurements the results of the fuzzy calculations are also not much different, the difference is the accuracy of the sensors in measuring existing goods. So in my opinion fuzzy logic here can produce output correctly because if the trash can is empty or filled a little then it is included in the small category, meaning that fuzzy logic here has succeeded in determining the value of the existing obscurity.

For the results of testing manual calculations and Arduino, there are differences due to rounding problems in the calculations. I will take an example with a height of 125mm and a weight of 320g, I found the results Z1 manual = 16.8, Z1 arduino = 16.67, Z3 manual = 23.2 Z3 arduino = 23.33. An example is in the image below.

```
******************************************************************************
||   Distance    ||   Weight   ||   Output   ||   Persen    ||   Capacity   ||
******************************************************************************
Tinggi   = 0.58  Cukup    = 0.42  Rendah   = 0.00
Ringan   = 1.00  Sedang    = 0.00 Berat   = 0.00
Sedikit   = 0.42 Lumayan    = 0.00        Penuh   = 0.58
A1 = 337.50      A2 = 0.00        A3 = 1865.50
Z1 = 16.67       Z2 = 0.00        Z3 = 23.33
defuzzify1 = 55.08
```

Illustration 5.2 : image testing result 1

So I will compare step by step calculating manually and Arduino. Object weighing 320 grams and height 135mm.

```
||   75 mm        || 320 g   ||     65.02    ||    69.00%    ||      Penuh     ||
Tinggi    = 0.75  Cukup    = 0.25  Rendah   = 0.00
Ringan    = 1.00  Sedang    = 0.00 Berat   = 0.00
Sedikit   = 0.25 Lumayan    = 0.00        Penuh   = 0.75
A1 = 202.50      A2 = 0.00        A3 = 2398.50
Z1 = 10.00       Z2 = 0.00        Z3 = 30.00
defuzzify1 = 65.02
```

Illustration 5.3 : image testing result 2

And the result is exactly the one on page 39. The A, Z values are the same and the final deffuzi result is the same. The second case i will try with weighing 320 grams and height 135mm.

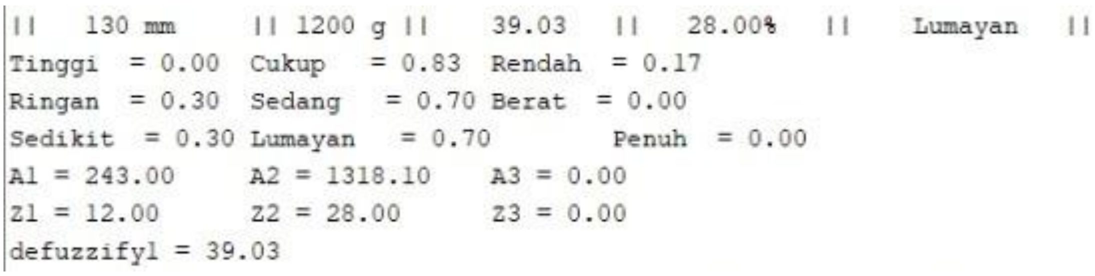
```
80mm 1200g
Sedikit  = 0.30 Lumayan   = 0.70

810x0.30 + 1883x0.70
40x0.30+ 40x 0.70

243 + 1318.1
40

39.02
```

Illustration 5.4 : image manual testing result 3



```
||   130 mm     || 1200 g ||    39.03    ||    28.00%    ||    Lumayan   ||
Tinggi  = 0.00  Cukup   = 0.83  Rendah  = 0.17
Ringan  = 0.30  Sedang   = 0.70 Berat  = 0.00
Sedikit  = 0.30 Lumayan   = 0.70        Penuh  = 0.00
A1 = 243.00     A2 = 1318.10    A3 = 0.00
Z1 = 12.00      Z2 = 28.00      Z3 = 0.00
defuzzifyl = 39.03
```

Illustration 5.5 : image arduino testing result 4

And it can be seen that the two numbers after the comma are quite influential in this calculation.

Berat asli (gr)	Berat hasil kalibrasi	Set scale
491 g	567 g	480
170 g	74 g	200
263 g	158 g	320
639 g	827 g	500
207 g	128 g	250

Table 5.5 : Table result of calibration testing

And the last test I tried was adjusting the weight calibration. According to the table above, the results can be said to be somewhat ambiguous and 480 in my opinion is the closest to the value of the praise. The results from the table above can also change if the pressure I put when placing things is different. So that the quality of the sensor used will play an important role in making this kind of tool.