Listing Program

```
// initialize the library with the numbers of the interface pins
#include <Servo.h>
#include <LiquidCrystal_I2C.h>
#include <Wire.h>
#include <Adafruit_PWMServoDriver.h>

Adafruit_PWMServoDriver pwm = Adafruit_PWMServoDriver();
#define MIN_PULSE_WIDTH     544
#define MAX_PULSE_WIDTH     2400
#define DEFAULT_PULSE_WIDTH 1500
#define FREQUENCY           50

uint8_t servonum0 = 0;
uint8_t servonum1 = 2;
uint8_t servonum2 = 4;
uint8_t servonum3 = 6;
uint8_t servonum4 = 8;
Servo servo1;
Servo servo2;

LiquidCrystal_I2C lcd(0x27, 16, 2);
int trigPin = 8;
int echoPin = 9;
int trigPin2 = 10;
int echoPin2 = 11;

void setup() {
  delay(1000);
  Serial.begin (9600);
  lcd.begin();
  lcd.setCursor(0,0);
  lcd.print("Jarak=");
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(trigPin2, OUTPUT);
  pinMode(echoPin2, INPUT);
  servo1.attach(4);
  servo2.attach(5);
  pwm.begin();
  pwm.setPWMFreq(FREQUENCY);
  delay(1000);
}

void loop() {
int j ;
  int duration, distance;
  digitalWrite (trigPin, HIGH);
  delayMicroseconds (10);
  digitalWrite (trigPin, LOW);
  duration = pulseIn (echoPin, HIGH);
  distance = (duration/2) / 29.1;
  int duration2, distance2;
        digitalWrite (trigPin2, HIGH);
        delayMicroseconds (10);
        digitalWrite (trigPin2, LOW);
        duration = pulseIn (echoPin2, HIGH);
        distance2 = (duration/2) / 29.1;

  lcd.setCursor(8,0);
  lcd.print(distance);
  if (distance >=8 && distance <=10){
    lcd.setCursor(11,0);
    lcd.print("CM");
    pwm.setPWM(servonum0, 0, pulseWidth(10));
    delay(1000);
    servo1.write(160); //Gripper buka
    servo2.write(42);
    delay(3000);
    pwm.setPWM(servonum1, 0, pulseWidth(30));
    delay(2000);
    servo1.write(42);
    servo2.write(160);
    delay(4000);
    pwm.setPWM(servonum1, 0, pulseWidth(10));
    delay(1000);
    pwm.setPWM(servonum0, 0, pulseWidth(80));
    delay(3000);
    pwm.setPWM(servonum1, 0, pulseWidth(20));
    delay(1000);
    servo1.write(160); //Gripper buka
    servo2.write(42);
    delay(4000);
    pwm.setPWM(servonum1, 0, pulseWidth(0));
    delay(1000);
  }

if (distance <=7){
    lcd.setCursor(11,0);
    lcd.print("MM");
    pwm.setPWM(servonum0, 0, pulseWidth(10));
  delay(1000);
  servo1.write(160); //Gripper buka
  servo2.write(42);
  delay(3000);
  pwm.setPWM(servonum1, 0, pulseWidth(30));
  delay(2000);
  servo1.write(42);
  servo2.write(160);
  delay(4000);
  pwm.setPWM(servonum1, 0, pulseWidth(10));
  delay(1000);
  pwm.setPWM(servonum0, 0, pulseWidth(90));
  pwm.setPWM(servonum0, 0, pulseWidth(130));
  pwm.setPWM(servonum0, 0, pulseWidth(160));
  delay(3000);
  pwm.setPWM(servonum1, 0, pulseWidth(20));
  delay(1000);
  servo1.write(160); //Gripper buka
  servo2.write(42);
  delay(4000);
  pwm.setPWM(servonum1, 0, pulseWidth(0));
  delay(1000);
}

if (distance >=16 ){
    lcd.setCursor(11,0);
lcd.print("CM");
  pwm.setPWM(servonum0, 0, pulseWidth(90));
  delay(3000);
}
```

```
}

int pulseWidth(int angle){
  int pulse_wide, analog_value;
  pulse_wide    = map(angle, 0, 180, MIN_PULSE_WIDTH, MAX_PULSE_WIDTH);
  analog_value  = int(float(pulse_wide) / 1000000 * FREQUENCY * 4096);
  Serial.println(analog_value);
  return analog_value;
}
```