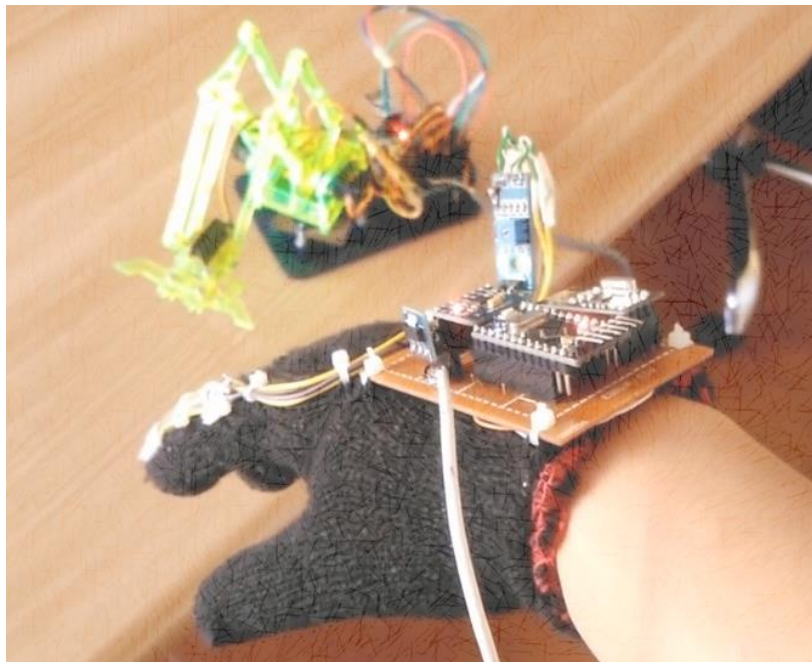


BAB V

IMPLEMENTASI DAN PENGUJIAN SISTEM

5.1. HASIL IMPLEMENTASI

Pada tahap ini penulis mengimplementasikan hasil rancangan yang telah dibuat pada tahap Implementasi yang dimaksud adalah proses menterjemahkan rancangan menjadi *software* dan berupa bentuk fisik robot. Adapun hasil implementasi tersebut adalah sebagai berikut :



Gambar 5.1 Kendali Robot Lengan

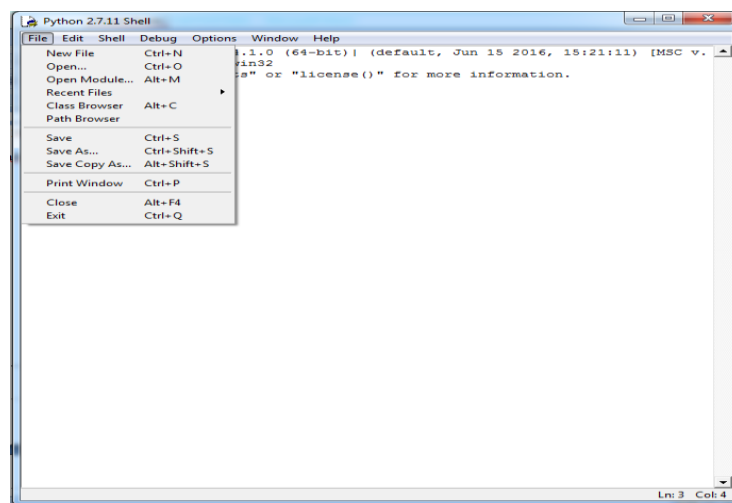
Gambar 5.1 merupakan hasil rancangan kendali lengan robot yang telah dirancang oleh penulis. Terdapat satu motor servo dipaling bawah untuk memutar kiri dan kekanan, dua servo ditengah berfungsi untuk maju mundur dan atas bawah dan satu servo didepan difungsikan untuk digunakan sebagai *gripper* (penjepit atau pemegang objek). Kotak belakang difungsikan untuk *casing*

Raspberry pi, Driver Motor Servo PWM PCA9685 12C, adaptor dan komponen lain.

5.2. PENGUJIAN PERANGKAT LUNAK

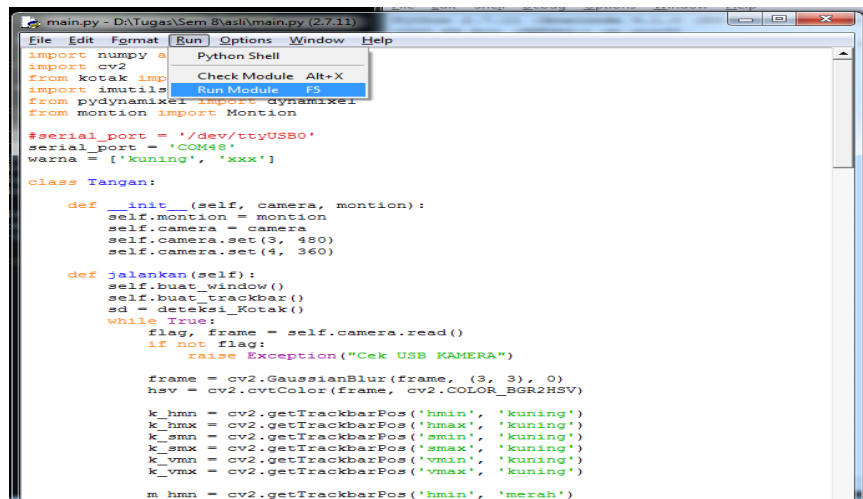
5.2.1. Python Idle

Untuk bahasa pemrograman python bisa menggunakan IDE *software* python idle, *software* python idle ini digunakan untuk membuat *listing* program dan menjalankan sekaligus program yang ditulis dengan bahasa pemrograman python. Pengujian *software* berikutnya yaitu pembuatan *listing* program dengan cara klik file pilith *new file* atau dengan menekan tombol ctrl+n. maka dapat dilihat seperti gambar 5.1



Gambar 5.1 Menu Membuat Program Baru

Setelah program yang kita buat sudah selesai untuk menjalankan program dengan cara berikut, dapat dilihat pada gambar 5.2 :



```

main.py - D:\Tugas\Sem 8\asli\main.py (2.7.11)
File Edit Format Run Options Window Help
Python Shell
Check Module Alt-X
Run Module F5
import numpy as np
import cv2
from kotak import *
import imutils
from pydynamixel import dynamixel
from montion import Montion

#serial_port = '/dev/ttyUSB0'
serial_port = 'COM48'
warna = ['kuning', 'xxx']

class Tangan:
    def __init__(self, camera, montion):
        self.montion = montion
        self.camera = camera
        self.camera.set(3, 480)
        self.camera.set(4, 360)

    def jalankan(self):
        self.buat_window()
        self.buat_trackbar()
        sd = deteksi_Kotak()
        while True:
            flag, frame = self.camera.read()
            if not flag:
                raise Exception("Cek USB KAMERA")

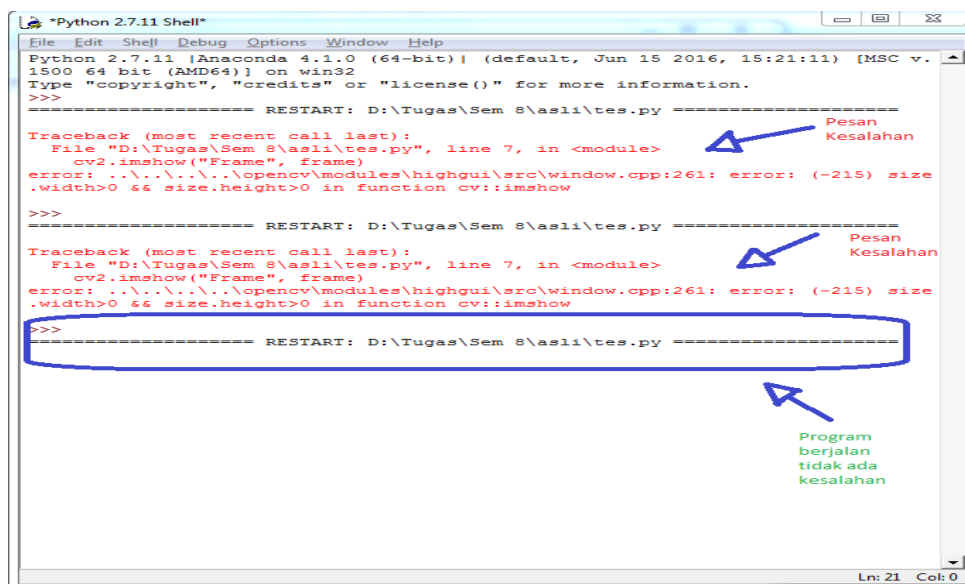
            frame = cv2.GaussianBlur(frame, (3, 3), 0)
            hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)

            k_hmn = cv2.getTrackbarPos('hmin', 'kuning')
            k_hmx = cv2.getTrackbarPos('hmax', 'kuning')
            k_smn = cv2.getTrackbarPos('smin', 'kuning')
            k_smx = cv2.getTrackbarPos('smax', 'kuning')
            k_vmn = cv2.getTrackbarPos('vmin', 'kuning')
            k_vmx = cv2.getTrackbarPos('vmx', 'kuning')
            m_hmn = cv2.getTrackbarPos('hmin', 'merah')

```

Gambar 5.2 Menjalankan Program Python

Setelah dijalankan akan ada pesan kelalahan jika terdapat kesalahan penulisan *sintax* jika tidak ada kesalahan maka program langsung berjalan seperti gambar, 5.2



```

Python 2.7.11 Shell
File Edit Shell Debug Options Window Help
Python 2.7.11 [Anaconda 4.1.0 (64-bit)] (default, Jun 15 2016, 15:21:11) [MSC v.
1500 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:\Tugas\Sem 8\asli\tes.py =====
Traceback (most recent call last):
  File "D:\Tugas\Sem 8\asli\tes.py", line 7, in <module>
    cv2.imshow("Frame", frame)
error: ..\..\..\opencv\modules\highgui\src>window.cpp:261: error: (-215) size
.width>0 && size.height>0 in function cv::imshow
>>>
===== RESTART: D:\Tugas\Sem 8\asli\tes.py =====
Traceback (most recent call last):
  File "D:\Tugas\Sem 8\asli\tes.py", line 7, in <module>
    cv2.imshow("Frame", frame)
error: ..\..\..\opencv\modules\highgui\src>window.cpp:261: error: (-215) size
.width>0 && size.height>0 in function cv::imshow
>>>
===== RESTART: D:\Tugas\Sem 8\asli\tes.py =====
>>>

```

Gambar 5.3 Output Program

5.2.2. Arduino IDE

Tahap ini meliputi pembahasan pada penulisan instruksi-instruksi program yang digunakan sistem secara keseluruhan. Untuk mendeteksi kemiringan yang terjadi pada tripod digunakan sensor MPU-6050, seperti yang sudah dijelaskan sebelumnya. Sensor akan mengirimkan data berupa nilai yaw, pitch dan roll menuju mikrokontroler. Pada proses pemrogramannya terdapat 2 tahap yang dilakukan.

Pertama, pengujian program kalibrasi sensor untuk menampilkan nilai rotasi matriks. Berikut adalah potongan sketch program kalibrasi sensor MPU-6050.

```

MPU6050_DMP6
/

// I2Cdev and MPU6050 must be installed as libraries, or else the .cpp/.h files
// for both classes must be in the include path of your project
#include "I2Cdev.h"

#include "MPU6050_6Axis_MotionApps20.h"
//#include "MPU6050.h" // not necessary if using MotionApps include file

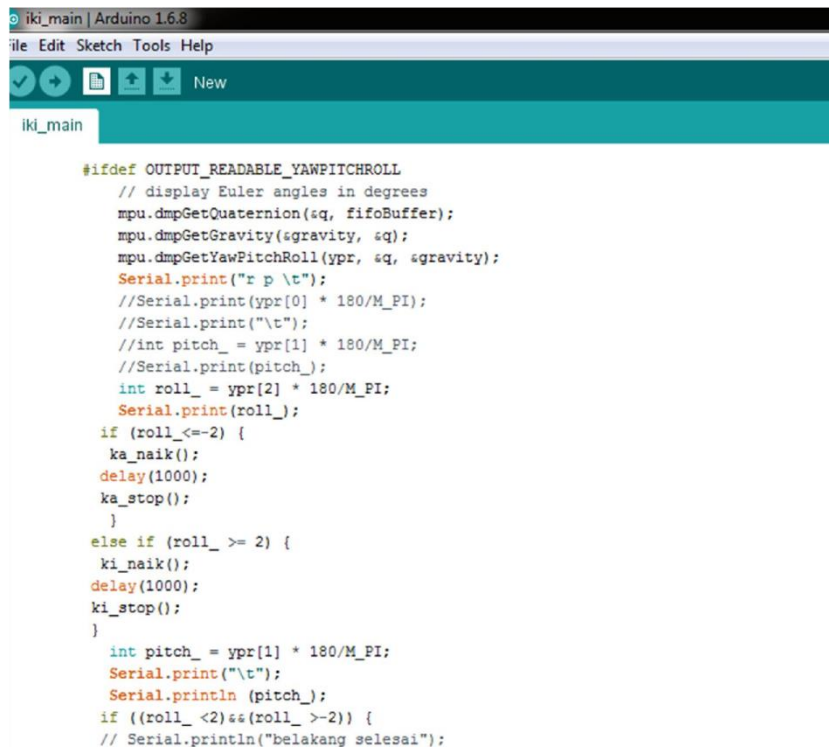
// Arduino Wire library is required if I2Cdev I2CDEV_ARDUINO_WIRE implementation
// is used in I2Cdev.h
#if I2CDEV_IMPLEMENTATION == I2CDEV_ARDUINO_WIRE
    #include "Wire.h"
#endif

// class default I2C address is 0x68
// specific I2C addresses may be passed as a parameter here
// AD0 low = 0x68 (default for SparkFun breakout and InvenSense evaluation board)
// AD0 high = 0x69
MPU6050 mpu;
//MPU6050 mpu(0x69); // <-- use for AD0 high

```

Gambar 4.7 Sketch Program Kalibrasi MPU-6050

Tahap kedua, menuliskan sketch tambahan ke dalam sketch program awal pengkalibrasian sensor MPU-6050. Berikut adalah potongan sketch programnya.



```

iki_main | Arduino 1.6.8
File Edit Sketch Tools Help
New
iki_main

#ifdef OUTPUT_READABLE_YAWPITCHROLL
  // display Euler angles in degrees
  mpu.dmpGetQuaternion(sq, fifoBuffer);
  mpu.dmpGetGravity(sgravity, sq);
  mpu.dmpGetYawPitchRoll(ypr, sq, sgravity);
  Serial.print("r p \t");
  //Serial.print(ypr[0] * 180/M_PI);
  //Serial.print("\t");
  //int pitch_ = ypr[1] * 180/M_PI;
  //Serial.print(pitch_);
  int roll_ = ypr[2] * 180/M_PI;
  Serial.print(roll_);
  if (roll_ <=-2) {
    ka_naik();
    delay(1000);
    ka_stop();
  }
  else if (roll_ >= 2) {
    ki_naik();
    delay(1000);
    ki_stop();
  }
  int pitch_ = ypr[1] * 180/M_PI;
  Serial.print("\t");
  Serial.println (pitch_);
  if ((roll_ <2)&&(roll_ >-2)) {
    // Serial.println("belakang selesai");
  }
}

```

Gambar 4.8 Sketch Program Pengoperasian Motor

5.3. PENGUJIAN PERANGKAT KERAS

Pengujian perangkat keras ini dilakukan untuk mengetahui benar atau tidaknya sebuah rangkaian listrik yang telah dirangkai. Pengujian dilakukan secara satu-persatu dari beberapa rangkaian yang telah selesai dibuat dan dengan alat bantu multimeter.

5.3.1. Pengujian Tegangan Pada Masing-masing Rangkaian

Pengujian tegangan yang dilakukan bertujuan untuk mengetahui seberapa besar tegangan pada setiap alat secara terpisah. Hal ini perlu diperhatikan karena beberapa komponen mempunyai tegangan yang berbeda disetiap rangkaian. Pengujian tegangan pada setiap rangkaian dapat dilihat pada tabel 5.1:

Tabel 5.1 Pengujian Tegangan

NO	Blok Rangkaian	Tegangan Yang diinginkan	Tegangan Sebenarnya
1	Regulator	12 volt	11,8 volt
2	Raspberry pi	5 volt	4.5 volt
3	Arduino Uno	5 volt	4.5 volt
4	Sensor mpu6050	5 volt	4.5 volt

5.3.2. Pengujian Motor Servo

Pengujian ini dilakukan untuk mengetahui sejauh mana lengan robot dapat dengan sesuai yang diinginkan. Hasil pengujian yang telah dilakukan dapat dilihat dalam tabel 5.2 berikut:

Tabel 5.2 Pengujian Motor Servo

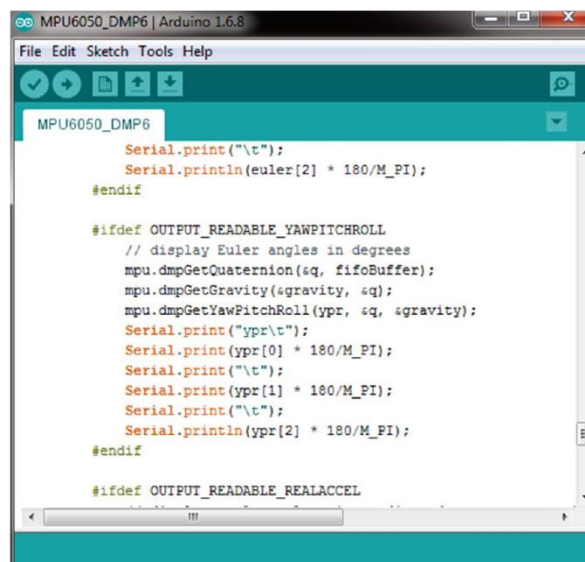
No	Jarak (cm)	ID Motor			
		A1	A2	A3	A4
1	13	90	133	33	90
2	16	90	100	45	90
3	24	90	85	60	90
4	30	90	70	60	90

5.3.3 Pengujian Sensor MPU-6050

Sensor MPU-6050, seperti yang sudah dijabarkan dalam bab sebelumnya, merupakan sebuah sensor kombinasi accelerometer dan gyroscope yang fungsinya mendeteksi percepatan gravitasi. Nilai perubahan percepatan gravitasi yang digunakan dalam sistem ini merupakan nilai rotasi matriks berupa yaw, pitch dan roll.

Pengujian dilakukan dengan cara menggerak-gerakkan sensor ke arah dan sudut yang berbeda-beda agar dapat diamati perubahan nilai rotasi

matriksnya. Perubahan nilainya dapat diamati di serial monitor Arduino. Berikut adalah sketch program untuk pengujian sensor MPU-6050 dan tampilan serial monitornya.



```

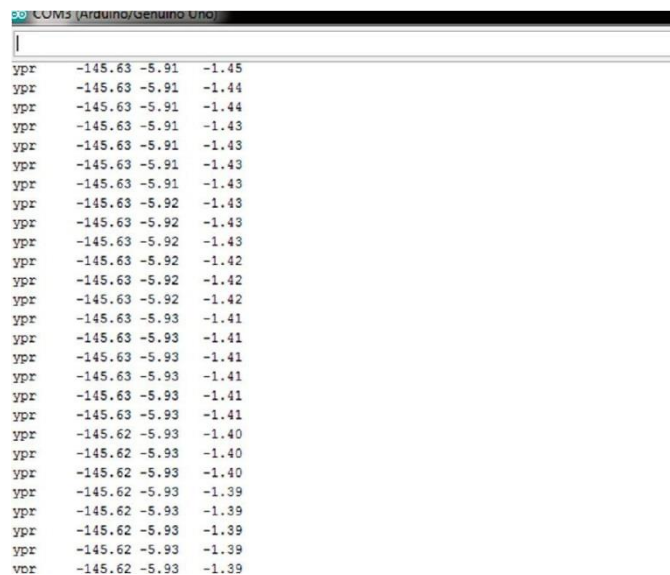
MPU6050_DMP6 | Arduino 1.6.8
File Edit Sketch Tools Help
MPU6050_DMP6
Serial.print("\t");
Serial.println(euler[2] * 180/M_PI);
#endif

#ifdef OUTPUT_READABLE_YAWPITCHROLL
// display Euler angles in degrees
mpu.dmpGetQuaternion(sq, fifoBuffer);
mpu.dmpGetGravity(sgravity, sq);
mpu.dmpGetYawPitchRoll(ypr, sq, sgravity);
Serial.print("ypr\t");
Serial.print(ypr[0] * 180/M_PI);
Serial.print("\t");
Serial.print(ypr[1] * 180/M_PI);
Serial.print("\t");
Serial.println(ypr[2] * 180/M_PI);
#endif

#ifdef OUTPUT_READABLE_REALACCEL

```

Gambar 4.4 Sketch Program Pengujian MPU-6050



```

COM3 (Arduino/Genuino Uno)
|
ypr -145.63 -5.91 -1.45
ypr -145.63 -5.91 -1.44
ypr -145.63 -5.91 -1.44
ypr -145.63 -5.91 -1.43
ypr -145.63 -5.91 -1.43
ypr -145.63 -5.91 -1.43
ypr -145.63 -5.91 -1.43
ypr -145.63 -5.92 -1.43
ypr -145.63 -5.92 -1.43
ypr -145.63 -5.92 -1.43
ypr -145.63 -5.92 -1.42
ypr -145.63 -5.92 -1.42
ypr -145.63 -5.92 -1.42
ypr -145.63 -5.93 -1.41
ypr -145.63 -5.93 -1.41
ypr -145.63 -5.93 -1.41
ypr -145.63 -5.93 -1.41
ypr -145.63 -5.93 -1.41
ypr -145.62 -5.93 -1.40
ypr -145.62 -5.93 -1.40
ypr -145.62 -5.93 -1.39
ypr -145.62 -5.93 -1.39
ypr -145.62 -5.93 -1.39
ypr -145.62 -5.93 -1.39
ypr -145.62 -5.93 -1.39

```

Gambar 4.5 Tampilan Nilai Yaw, Pitch, dan Roll oleh MPU-6050

Dari hasil pengamatan di atas diketahui bahwa sensor MPU-6050 bekerja dengan baik. Ketika fisik sensor digerakkan dari satu arah ke arah yang lain,

nilai langsung berubah sesuai dengan pergerakannya. Namun, dibutuhkan waktu beberapa saat hingga nilai stabil di satu titik. Hal tersebut dikarenakan sensor MPU-6050 sangat sensitif terhadap getaran yang menyebabkan nilai mudah berubah apabila bergeser 1^o saja. Adanya getaran tersebut karena faktor manusia, seperti memegang sensor dengan tidak stabil, dan faktor lingkungan, seperti tertiup angin.

5.3.4. Pengujian *Flex Sensor*

Posisi sensor flex pada jari berdasarkan kepada dua kondisi yaitu pada saat membuka dan mengepal. Hasil pengujian terlihat pada tabel 6.

Tabel 5.3 Hasil Pengujian Sensor Flex Pada Jari

Percobaan	Kondisi	R (multimeter)	ADC	V Out	R(teori)
1	Membuka	30,28 k Ω	250	1,22	30,98 k Ω
	Mengepal	46,76 k Ω	180	0,87	47,47 k Ω
2	Membuka	32,35 k Ω	235	1,14	33,85 k Ω
	Mengepal	42,83 k Ω	190	0,92	44,34 k Ω
3	Membuka	34,38 k Ω	225	1,09	35,87 k Ω
	Mengepal	75,54 k Ω	120	0,58	76,20 k Ω
4	Membuka	28,30 k Ω	270	1,32	27,87 k Ω
	Mengepal	45,10 k Ω	180	0,87	47,47 k Ω
5	Membuka	27,55 k Ω	265	1,29	28,75 k Ω
	Mengepal	35,11 k Ω	220	1,07	36,72 k Ω

5.3.5 Pengujian Jarak dan Keadaan Real-Time

Pengujian ini dilakukan untuk mengetahui seberapa jauh batas maksimum jarak yang dibutuhkan agar antara sarung tangan sebagai input dan robot tangan sebagai output. Pengujian keadaan Real-Time ini bertujuan untuk mendapatkan seberapa cepat respon robot tangan ini ketika sarung tangan yang sebagai

masukan mengirimkan sebuah informasi adanya lekukan pada flex sensor. Sehingga dapat diketahui seberapa pengaruh hubungan antara jarak dan delay pada alat ini. Hasil pengujian jarak ditunjukkan pada tabel 3.

Tabel 5.4 Hasil Pengujian Jarak dan Keadaan Real Time

Percobaan	Jarak (m)	Hasi	Dela
1	3	OK	Sangat Kecil
2	6	OK	Sangat Kecil
3	9	OK	Sangat Kecil
4	12	OK	Keci
5	15	OK	Keci
6	18	OK	Keci
7	21	OK	Keci
8	24	OK	Sedang
9	27	Gaga	-

Pada pengujian ini perangkat yang penulis uji ialah modul wireless xbee. Pengujian ini bertujuan untuk mengetahui seberapa jauh cakupan modul Xbee S1 ini dan pergerakan Real-Time. Pada pengujian ini delay yang didapat sangatlah kecil sehingga sistem ini dapat dikatakan bergerak secara Real-Time. Pengujian ini dilakukan di dalam ruangan. Seperti yang tertera pada tabel hasil pengujian Xbee tidak selalu sesuai dengan spesifikasi yang tertera, hal ini disebabkan oleh banyaknya penghalang ataupun gangguan pada saat mengirimkan informasi. Pada pengujian ini juga kita dapat melihat bahwa jarak mempengaruhi delay sebuah sistem dimana semakin jauh jarak yang di uji maka semakin besar delay yang dihasilkan.

5.3.6 Pengujian Keseluruhan

Pengujian keseluruhan dilakukan untuk mengetahui apakah robot mampu beropersai sesuai dengan yang diinginkan, dengan kondisi jarak dan cahaya yang bervariasi. Hasil pengujian dapat dilihat pada tabel 5.5 berikut:

Tabel 5.5 Pengujian keseluruhan

No	Pengujian	Proses dan Output	Hasil	
			Ya	Tidak
1	Motor Servo	Putar Kiri-Kanan, Depan-Belakang, Atas-Bawah	V	
2	Sensor MPU6050	Lengan robot bisa bergerak mengikuti gerak Accelerometer Melalui Wireless nRF24L01	V	
3	Flex Sensor	Jari-jari pada servo 4 lengan robot bergerak sesuai kelengkungan Flex Sensor.	V	
4	Wireless nRF24L01	Wireless nRF24L01 mengirim perintah dari Flex Sensor dan Accelerometer untuk menggerakkan lengan robot yang diproses pada arduino.	V	

Dari tabel hasil pengujian 5.5 dapat diambil kesimpulan sistem penggerak arah robot akan bekerja setelah menerima perintah dari Handphone Android melalui media transmisi Bluetooth yang di proses di arduino yang terdapat pada Robot. Servo pada jari – jari robot akan bergerak sesuai dengan lengkukan yang di terapkan pada Flex sensor begitu juga lengan robot akan bergerak sesuai arah dari Accelerometer sensor, perintah – perintah ini akan di proses menggunakan perangkat Arduino yang ada pada sarung tangan pengendali dan akan di kirimkan ke robot menggunakan media transmisi wireless pada modul nRF24L01 dan akan diterima pada perangkat robot.

5.4 ANALISA SISTEM SECARA KESELURUHAN

Untuk mendeteksi apabila terjadi kesalahan setelah uji coba, maka perlu dilakukan analisa rangkaian secara keseluruhan. Dari seluruh proses yang telah dilakukan, baik pengujian perangkat keras maupun perangkat lunak, dapat dikatakan bahwa alat ini dapat berfungsi sebagaimana yang penulis inginkan. Proses pengenalan bentuk pun tidak terjadi kesalahan pembacaan data, motor servo dapat berputar sesuai program yang di buat penulis, dan mampukah robot mengangkat beban.

Sistem kendali robot nirkabel akan terjadi kesalahan mengirim nilai data apabila melewati jarak yang ditentukan. Untuk mengatasi pengiriman nilai agar tidak terjadi kesalahan maka jarak antara kendali dan robot harus sekitar 5 – 10 m.

Pengujian dilakukan untuk menunjukkan bahwa robot ini dapat bekerja sesuai dengan tujuan dari pembuatan. Pengujian ini dilakukan dengan cara sebagai berikut :

1. Sistem kendali gerakan tangan akan diproses kemudian akan dikirim ke Raspberry Pi dan di cocok kan dengan gerakan yang pada robot.
2. Apabila jarak kendali terlalu jauh dari jangkauan, maka robot tidak bisa bergerak mengikuti gerakan tangan. Sebaiknya robot dan kendali diletakkan dalam jarak optimal.