

## **BAB IV**

### **PENGUJIAN ALAT DAN PEMBAHASAN**

#### **4.1 Uji Coba Alat**

Dalam bab ini akan dibahas mengenai pengujian alat yang telah dibuat. Dimulai dengan pengujian setiap bagian-bagian dari hardware dan software yang bertujuan untuk mengetahui apakah setiap bagiannya bekerja dengan baik. Dilanjutkan dengan pengujian sistem secara keseluruhan yang bertujuan untuk mengetahui apakah antarbagian sistem bekerja dengan baik sehingga bisa menghasilkan satu sistem utuh sesuai yang diharapkan serta melakukan pendataan pada sistem tersebut.

#### **4.2 Prinsip Kerja Alat**

*Prototype* sistem *self-leveling* pada kaki tripod ini dibuat untuk memberikan gambaran rancangan penstabil tripod dengan pengaturan kaki tripod yang otomatis. Sistem ini membantu pengguna kamera untuk tidak melakukan pengaturan secara manual. Berdasarkan uraian perancangan pada bab sebelumnya, diharapkan sistem ini dapat digunakan sesuai dengan tujuannya.

Sistem *self-leveling* pada kaki tripod akan bekerja (ON) saat sistem mendapat suplai tegangan dari rangkaian catu daya yang tersambung dengan sumber PLN. Rangkaian catu daya sebesar 12V tersebut akan memberikan suplai ke semua bagian komponen mulai dari sensor MPU-6050, yang mendeteksi kemiringan pada tripod, dan motor akan berputar apabila motor driver diaktifkan sesuai dengan setting mikrokontroler berdasarkan input nilai kemiringan yang terdeteksi sensor. Tetapi apabila sensor sudah dalam keadaan seimbang (nilai kemiringan 0), maka motor driver tidak aktif dan otomatis motor tidak berputar. Bentuk fisik dari sistem *Self-Leveling* pada Tripod Kamera Berbasis Mikrokontroler dapat dilihat pada gambar berikut.



**Gambar 4.1** Bentuk Fisik Sistem *Self-Leveling* Tripod

Uraian pengoperasian sistem *self-leveling* tripod ini adalah sebagai berikut:

1. Kabel catu daya tripod dihubungkan ke sumber PLN 220V.
2. Sensor MPU-6050 akan mendeteksi berapa nilai kemiringan (*pitch* dan *roll*) pada ketiga kaki tripod.
3. Motor driver akan aktif apabila nilai *pitch* dan *roll* tidak berada dalam range seimbang yang disetting, kemudian motor driver menyalakan motor agar berputar ke kanan atau ke kiri.
4. Motor akan berhenti berputar apabila kedudukan tripod sudah dinyatakan seimbang sesuai dengan program yang disetting (nilai kemiringan 0).

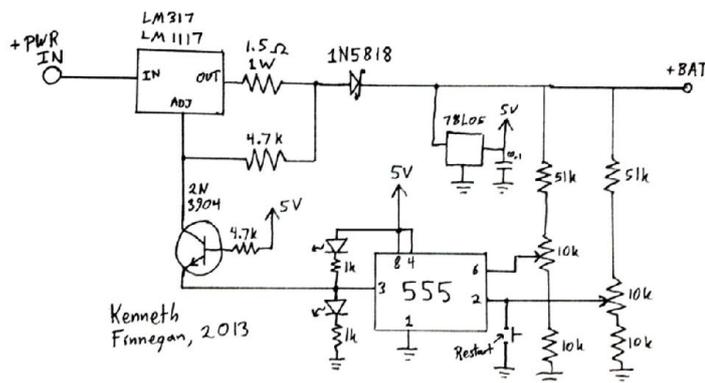
### **4.3 Pengujian Hardware**

#### **4.3.1 Pengujian Catu Daya**

Catu daya berguna untuk memberikan tegangan yang stabil ke seluruh rangkaian sistem. Dalam sistem self-leveling kaki tripod ini digunakan catu daya sebesar 12VDC yang dapat diambil dari rangkaian baterai, Adaptor, maupun *Power Supply*.

Dalam percobaan sistem, catu daya yang dipakai berasal dari Adaptor dan *Power Supply* yang sudah disetting 12VDC karena dianggap memudahkan dalam pengujiannya. Kekurangannya, catu daya macam itu penggunaannya membutuhkan sumber tegangan PLN. Apabila pengujian tripod dilakukan diluar ruangan tentu hal tersebut akan menyulitkan.

Catu daya yang berasal dari baterai dibuat dengan merakit 4 buah baterai isi ulang 3,7V yang tersambung dengan rangkaian untuk *auto-charge* baterai seperti gambar di bawah.



**Gambar 4.2** *Auto-Charge Battery Circuit*

Setelah itu rangkaian dites menggunakan multimeter dan menunjukkan tampilan +15VDC

### 4.3.2 Pengujian Sistem Arduino UNO R3

Pengujian sistem ini dilakukan dengan cara memberikan suplai tegangan terhadap rangkaian Arduino kemudian mengecek fungsi bagian-bagian seperti port-port I/O dan tombol reset, dan lampu indikator. Sistem Arduino UNO R3 disuplai dengan catu daya

sebesar 12VDC yang dialirkan melalui *switching regulator* sehingga menghasilkan 5VDC. Sistem Arduino akan dikatakan baik apabila ketika Arduino diinputkan program dari *software*, multimeter dapat menampilkan nilai 5VDC, untuk logika “HIGH” dan nilai 0VDC untuk logika “LOW”.



```
sketch_jun06a | Arduino 1.6.8
File Edit Sketch Tools Help
sketch_jun06a $
void setup() {
  //
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);
  pinMode(7, OUTPUT);
  pinMode(8, OUTPUT);
  pinMode(9, OUTPUT);
}

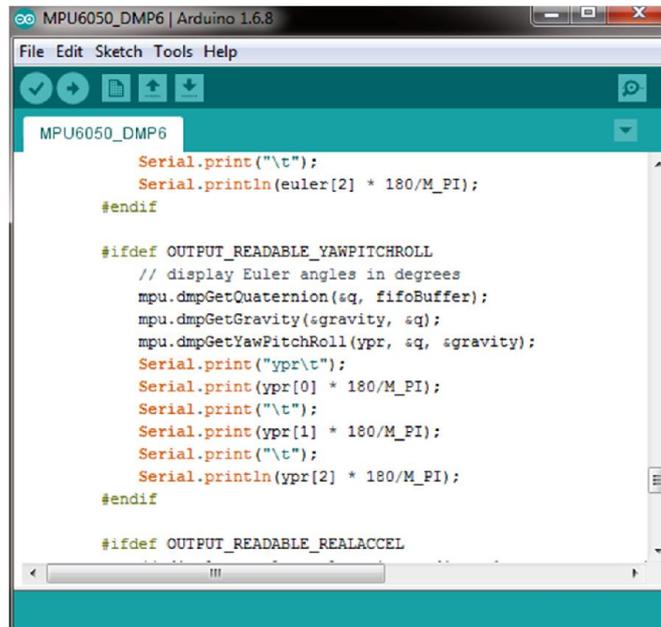
void loop() {
  //
  digitalWrite(5, HIGH);
  digitalWrite(6, LOW);
  digitalWrite(7, LOW);
  digitalWrite(8, HIGH);
  digitalWrite(9, HIGH);
}
```

**Gambar 4.3** Sketch Program Pengujian Arduino

### 4.3.3 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.

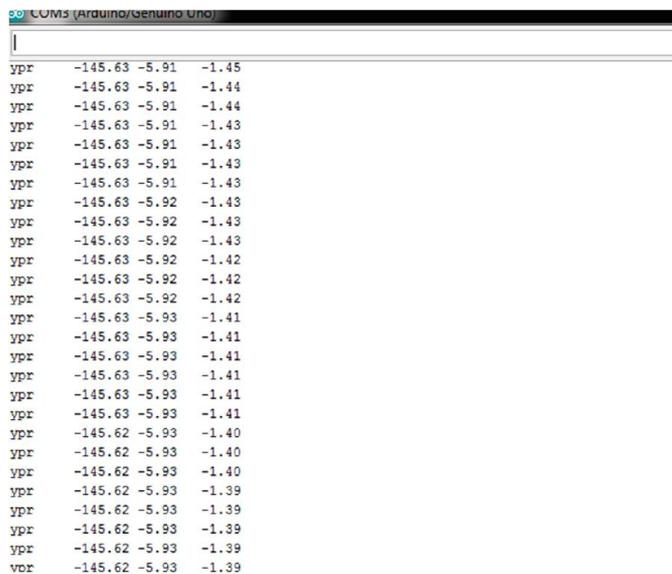


```
MPU6050_DMP6 | Arduino 1.6.8
File Edit Sketch Tools Help
MPU6050_DMP6
Serial.print("\t");
Serial.println(euler[2] * 180/M_PI);
#endif

#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 -145.62 -5.93 -1.40
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
```

**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° saja. Adanya getaran tersebut karena faktor manusia, seperti memegang sensor dengan tidak stabil, dan faktor lingkungan, seperti tertiup angin.

#### **4.3.4 Pengujian Motor Driver L298N**

Pada pengujian *motor driver* L298N untuk rangkaian sistem ini dilakukan dengan mengecek menggunakan multimeter pada output masing-masing driver. Dimulai dengan menghubungkan motor driver ke Arduino. Input dari Arduino berupa logika “HIGH” dan “LOW”, berisi perintah agar motor melakukan perputaran ke kiri dan ke kanan yang dimasukkan secara bergantian ke dalam motor driver. Motor driver dinyatakan baik apabila output motor driver menghasilkan tegangan +12VDC dan -12VDC secara bergantian ketika dilakukan pengecekan menggunakan multimeter.

Sebagai acuannya, output yang menghasilkan tegangan +12VDC menandakan driver mengaktifkan motor untuk berputar ke kanan. Dan output dengan tegangan -12VDC mengaktifkan motor untuk berputar ke kiri.

#### **4.3.5 Pengujian Kemiringan Terhadap Tripod**

Pengujian kemiringan terhadap tripod dilakukan jika kesatuan sistem sudah dirangkai, dimulai dari sensor MPU-6050 yang terhubung dengan input Arduino UNO R3, kemudian output-nya terhubung dengan input motor driver, dan output motor driver dihubungkan ke motor di kaki-kaki tripod. Pengujian dilakukan dengan cara memperhatikan perubahan nilai *pitch* dan *roll* di serial monitor, lalu mengamati perputaran motor setiap kaki tripodnya.

Langkah awal yang dilakukan adalah *setting* tripod agar berada di posisi seimbang. Dikatakan seimbang apabila sensor pada perangkat elektronik yang terpasang di tripod menunjukkan nilai *pitch* dan *roll* berada di dalam lingkup nilai toleransi.



	kiri)												
Kaki belakang	Kaki depan	-1	-1	-2	9	-2	21	-2	24	-2	30	--	--
Kaki depan & kanan	Kaki kiri	-1	-1	-10	-7	-17	-11	-24	-14	-35	-20	--	--
Kaki depan & kiri	Kaki kanan	1	1	8	-7	15	-10	24	-14	35	-19	--	--

Dari tabel di atas diperoleh bahwa besar sudut kemiringan yang mampu ditanggung oleh tripod kurang lebih berada di  $0^\circ$  sampai  $20^\circ$ . Percobaan dilakukan dengan tripod dalam keadaan seimbang ( $0^\circ$ ) yang kemudian dimiringkan di tiap sisinya hingga batas kemiringan kritis, yaitu  $20^\circ$  sebelum tripod akhirnya jatuh karena tidak kuat menopang. Sedangkan waktu untuk tripod supaya bergerak dari keadaan tidak seimbang menuju seimbang dibutuhkan antara 5 sampai dengan 10 detik untuk tiap perubahan nilai +/-1, jadi misalkan kaki tripod mengalami kemiringan di nilai pitch sebesar 5 agar kembali ke nilai toleransi yaitu 2 maka estimasi waktu yang dibutuhkan sekitar 15 sampai 30 detik.

Adapun nilai error yang terjadi saat pengukuran nilai kemiringan oleh sistem ini ditentukan oleh jenis sensor accelerometer yang dipakai. Untuk jenis sensor MPU-6050 tertera pada *datasheet*-nya sebagai *Initial Calibration Tolerance* atau nilai toleransi kalibrasi saat pengukuran adalah sebesar  $\pm 3\%$ .

Selain itu, permasalahan juga terletak pada panjang baut yang dipasang pada modifikasi motor DC. Baut yang disediakan hanya sepanjang 25cm, sedangkan untuk menyesuaikan otomatisasi kaki tripod dibutuhkan lebih dari 25cm.

#### 4.4 Pengujian Software

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"

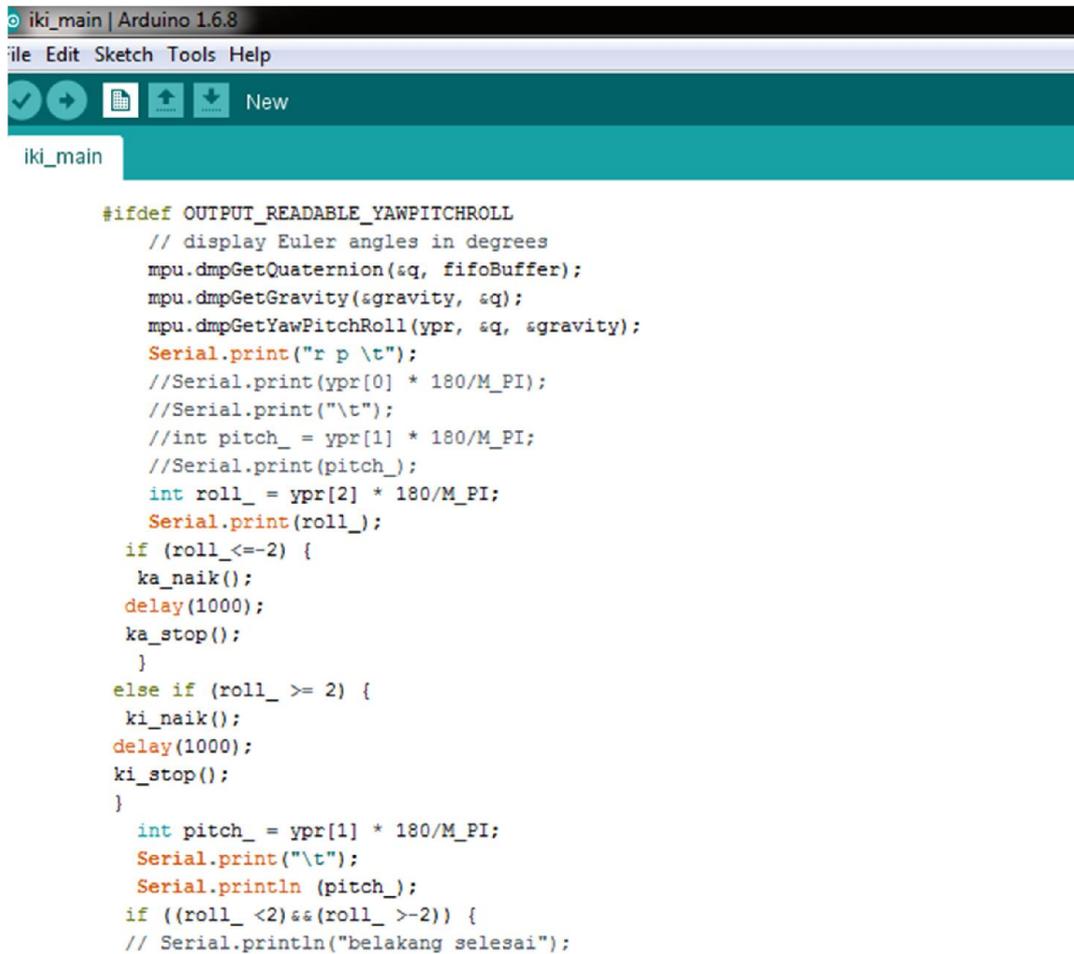
#include "MPU6050_6Axis_MotionApps20.h"
//#include "MPU6050.h" // not necessary if using MotionApps include file

// 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
iki_main

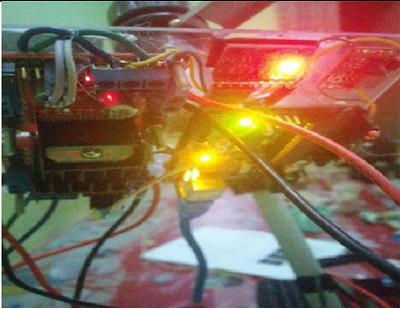
#ifdef OUTPUT_READABLE_YAWPITCHROLL
  // display Euler angles in degrees
  mpu.dmpGetQuaternion(&q, fifoBuffer);
  mpu.dmpGetGravity(&gravity, &q);
  mpu.dmpGetYawPitchRoll(ypr, &q, &gravity);
  Serial.print("r p \t");
  //Serial.print(ypr[0] * 180/M_PI);
  //Serial.print("\t");
  //int pitch_ = ypr[1] * 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");
  }
}
#endif
```

**Gambar 4.8** *Sketch* Program Pengoperasian Motor

#### 4.5 Pembahasan Secara Keseluruhan

Sistem *Self-Leveling* Tripod ini dirancang dan dibuat serta diuji sesuai dengan batasan-batasan yang telah ditentukan sebelumnya. Pengujiannya tidak terbatas berada di dalam atau di luar ruangan. Berikut adalah gambaran analisis secara keseluruhan dari kerja Sistem *Self-Leveling* Tripod.

**Tabel 4.2** Kerja Sistem Keseluruhan

No	Kondisi	Gambar	Keterangan
1.	Lampu indikator sensor, Arduino, dan motor driver menyala		Menandakan bahwa sumber catu daya disambungkan ke sumber tegangan PLN.
2.	Saat kondisi tripod masih diam	 kaki kanan tripod	Sensor sedang mendeteksi perubahan nilai-nilai (pitch dan roll) dan distabilkan.

			
3.	Saat motor mulai berputar	Kaki depan tripod	<p>Sesudah sensor menstabilkan nilai-nilai yang terdeteksi, Arduino akan mengaktifkan motor driver agar motor berputar.</p>

4.	Saat motor di kaki kanan/kiri tripod yang berputar		Saat motor di kaki kanan tripod berputar, itu artinya nilai roll mengalami perubahan ke angka positif. Apabila kaki kiri tripod yang berputar maka nilai roll mengalami perubahan ke angka negatif.
----	--	--	---

5.	Saat motor di kaki depan yang berputar		Saat motor di kaki depan tripod berputar, hal ini kemungkinan nilai pitch bisa mengalami perubahan ke angka positif atau negatif. Jika berubah ke angka positif maka motor berputar ke kiri dan jika berubah ke angka negatif maka motor berputar ke kanan.
6.	Saat motor berhenti berputar		Arduino membaca nilai-nilai yang dideteksi oleh sensor sudah berada di dalam batas toleransi sehingga menonaktifkan motor driver dan

			<p>motor pun berhenti.</p> <p>Bulatan merah pada gambar menunjukkan perubahan panjang baut dikarenakan proses self-leveling oleh program.</p>
--	--	--	---