

BAB IV

PENGUJIAN SISTEM

Tahapan pengujian sistem merupakan langkah yang dilaksanakan setelah proses perancangan atau perakitan alat selesai, dengan tujuan utama untuk memperoleh data percobaan yang diperlukan. Tahap ini bertujuan untuk memastikan bahwa alat tersebut berfungsi secara optimal sesuai dengan spesifikasi dan rancangan yang telah dibuat sebelumnya. Melalui serangkaian uji coba yang dilakukan secara berulang, pengujian bertujuan untuk mengidentifikasi performa alat, memverifikasi apakah alat tersebut bekerja sebagaimana mestinya, serta memastikan akurasi dan keandalan data yang dihasilkan. Beberapa kali percobaan dilakukan agar dapat mengatasi potensi kesalahan atau ketidaksesuaian, sehingga data yang diperoleh benar-benar mencerminkan kondisi alat dalam situasi sebenarnya. Dengan demikian, pengujian ini sangat penting dalam menjamin alat dapat berfungsi sesuai dengan kebutuhan dan harapan pengguna atau perancangannya.

4.1. Pengujian Komponen

4.1.1. Pengujian Catu Daya

Pengujian catu daya baterai bertujuan untuk memastikan bahwa adaptor memiliki tegangan yang cukup sebelum digunakan dalam sistem. Adaptor ini akan menjadi sumber daya utama untuk berbagai komponen, seperti mikrokontroler ESP8266, motor servo MG995, dan sensor-sensor pada perangkat tikus berbasis IoT. Untuk menguji adaptor, digunakan multimeter, yang diatur pada mode pengukuran tegangan DC.

Tabel 4. 1 Pengujian Adaptor Menggunakan Multimeter

No	Pengukuran	Tampilan Multimeter	Keterangan
1	Output		Output tegangan 5.73 VDC

Pada pengujian ini, digunakan alat ukur multimeter digital yang difungsikan untuk mengukur tegangan DC (*Direct Current*). Proses pengujian dilakukan dengan menghubungkan *probe* multimeter ke output adaptor, di mana *probe* merah dihubungkan ke kutub positif dan *probe* hitam ke kutub negatif. Hasil pengukuran yang ditampilkan pada layar multimeter menunjukkan nilai sebesar 5,73 VDC. Nilai ini menunjukkan bahwa adaptor mampu memberikan tegangan keluaran dalam kisaran yang stabil dan sesuai dengan kebutuhan kerja komponen utama dalam sistem, khususnya mikrokontroler ESP8266 yang bekerja pada tegangan 3,3V hingga 5V. Pengukuran ini ditampilkan dalam Tabel 4.1 sebagai bukti bahwa adaptor berfungsi dengan baik dan layak digunakan sebagai sumber daya pada sistem secara keseluruhan.

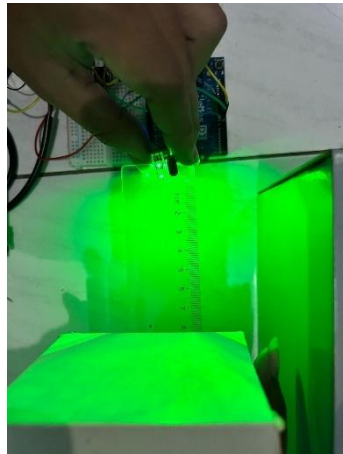
4.1.2. Pengujian Sensor *Infrared Obstacle*

Pengujian sensor *Infrared Obstacle* dilakukan untuk mengukur kemampuan sensor dalam mendeteksi jarak suatu objek, sekaligus memastikan bahwa sensor tersebut berfungsi dengan optimal. Proses ini penting untuk mengetahui apakah sensor *Infrared Obstacle* mampu mendeteksi objek dengan akurat sesuai dengan spesifikasinya, serta mengidentifikasi jika terdapat masalah atau malfungsi pada komponen sensor yang dapat mempengaruhi kinerjanya sebagai alat pendeteksi objek. Pada Tabel 4.2 adalah data yang diperoleh pada pengujian sensor *Infrared Obstacle*.

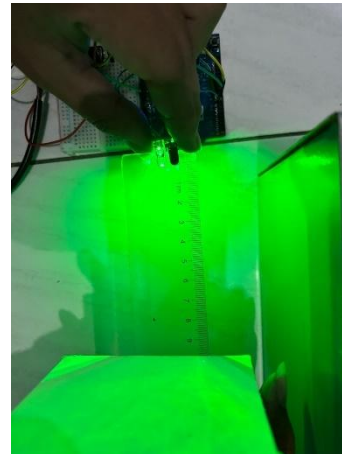
Tabel 4. 2 Pengujian Sensor *Infrared Obstacle*

Pengujian	Jarak sensor ke objek (cm)	Indikator lampu sensor infrared	Kondisi
1	8	Hidup	Terdeteksi
2	10	Hidup	Terdeteksi
3	12	Hidup	Terdeteksi
4	15	Mati	Tidak Terdereksi
5	20	Mati	Tidak Terdereksi

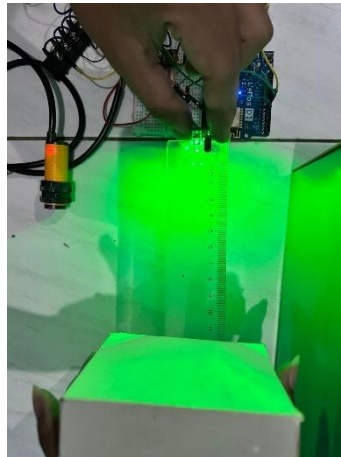
Berdasarkan Tabel 4.2 hasil pengujian yang dilakukan, sensor *Infrared Obstacle* terbukti berfungsi dengan baik sesuai dengan perannya sebagai pendeteksi objek. Sensor tersebut mampu mendeteksi objek dengan maksimal pada jarak hingga 12 cm. Namun, jika objek berada pada jarak yang lebih jauh dari 12 cm, sensor tidak lagi mampu mendeteksi keberadaan objek tersebut.



(a)



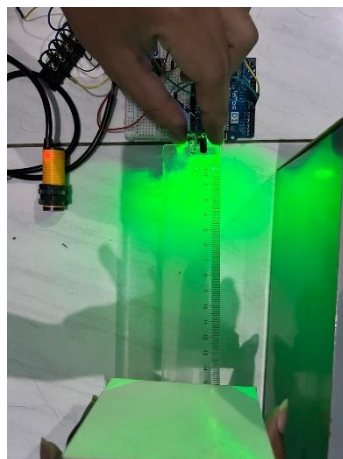
(b)



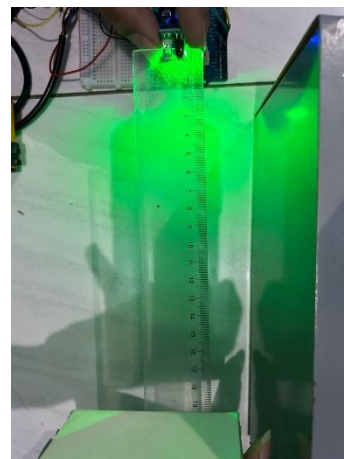
(c)

Gambar 4. 1 Pengujian Sensor *Infrared Obstacle* (a) 8cm, (b) 10cm, (c) 12cm

Dari hasil pengujian yang dilakukan pada Gambar 4.2 jarak 8cm, 10cm dan 12cm, sensor *Infrared Obstacle* mampu mendeteksi keberadaan objek. Deteksi ini menyebabkan lampu indikator menyala, karena sensor menerima pantulan sinar inframerah yang dipancarkan oleh pemancar. Sinar pantulan tersebut menandakan bahwa objek berada dalam jangkauan deteksi, sehingga sensor dapat berfungsi sebagaimana mestinya dan mengaktifkan lampu indikator sebagai respons terhadap adanya objek di depannya. Pengujian ini menegaskan bahwa sensor inframerah bekerja secara efektif pada jarak tersebut.



(d)



(e)

Gambar 4. 2 Pengujian Sensor *Infrared Obstacle* (d) 15 cm , (e) 20 cm

Dalam pengujian yang dilakukan pada jarak 15 cm dan 20 cm, sensor *Infrared Obstacle* tidak mampu mendeteksi keberadaan objek di depannya. Hal ini menyebabkan tidak adanya sinar inframerah yang terpantul kembali ke sensor, sehingga lampu indikator tidak menyala. Kondisi ini menunjukkan bahwa pada jarak tersebut, sensor berada di luar batas jangkauan deteksinya, seperti yang terlihat pada Gambar 4.2.

4.1.3. Pengujian Sensor *Proximity* E18-D80NK

Pengujian sensor *Proximity* E18-D80NK dilakukan untuk mengetahui apakah sensor *Proximity* E18-D80NK dapat mendeteksi objek dengan akurasi sesuai dengan spesifikasinya. Selain itu, pengujian ini juga bertujuan untuk mengidentifikasi adanya masalah atau malfungsi pada komponen sensor yang dapat memengaruhi kinerjanya sebagai alat pendeteksi objek. Pada Tabel 4.3 adalah data yang diperoleh pada pengujian sensor *Proximity* E18-D80NK.

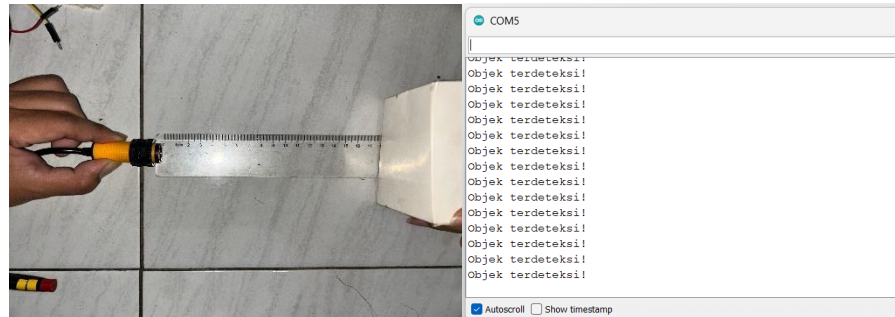
Tabel 4. 3 Pengujian Sensor *Proximity* E18-D80NK

No	Jarak Pengukuran	Indikator led sensor proximity	Kondisi
1	10 cm	Mati	Objek Terdeteksi
2	18 cm	Mati	Objek Terdeteksi
3	19 cm	Hidup	Tidak ada Objek
4	30 cm	Hidup	Tidak ada Objek

Berdasarkan Tabel 4.3 hasil pengujian yang dilakukan, sensor *Proximity* E18-D80NK terbukti berfungsi dengan baik sesuai dengan perannya sebagai pendeteksi objek. Hasil pengujian menunjukkan bahwa pada jarak 10 cm, indikator LED sensor dalam keadaan "Mati," yang menunjukkan bahwa objek terdeteksi. Begitu pula pada jarak 18 cm, indikator juga "Mati," yang menandakan objek terdeteksi. Namun, pada pengujian ketiga dengan jarak 19 cm, indikator berubah menjadi "Hidup," yang berarti tidak ada objek yang terdeteksi. Hal yang sama terjadi pada pengujian keempat dengan jarak 30 cm,

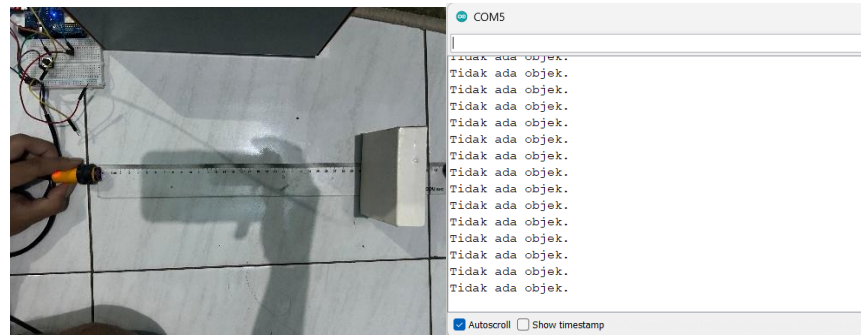
di mana indikator juga "Hidup," menunjukkan bahwa tidak ada objek yang terdeteksi.

Dari hasil pengujian ini, dapat disimpulkan bahwa sensor Proximity E18-D80NK berhasil mendeteksi objek pada jarak 10 cm dan 18 cm, sementara pada jarak 19 cm dan 30 cm, sensor tidak mampu mendeteksi objek. Hal ini menunjukkan batas deteksi sensor dalam kondisi yang diuji.



Gambar 4. 3 Pengujian Sensor Proximity 18 cm

Dari hasil pengujian yang dilakukan pada Gambar 4.4 jarak 18 cm, sensor *Proximity* E18-D80NK mampu mendeteksi keberadaan objek. Dengan menggunakan alat pengukur mistar sebagai acuan jarak ketika sensor mendeteksi adanya objek maka menyebabkan lampu indikator tidak menyala, karena sensor menerima pantulan sinar inframerah yang dipancarkan oleh pemancar. Sinar pantulan tersebut menandakan bahwa objek berada dalam jangkauan deteksi. Hasil dari pengujian ditampilkan melalui antarmuka pemrograman. Output menunjukkan bahwa sensor berhasil mendeteksi objek pada jarak yang telah ditentukan, dengan keluaran pesan berulang yang menyatakan "Objek terdeteksi!" Hal ini mengindikasikan bahwa sensor Proximity E18-D80NK berfungsi dengan baik dan mampu mengidentifikasi keberadaan objek secara efektif pada jarak yang diuji.



Gambar 4. 4 Pengujian Sensor Proximity 30 cm

Dari hasil pengujian yang dilakukan pada Gambar 4.3 jarak 30 cm, sensor *Proximity* E18-D80NK tidak mampu mendeteksi keberadaan objek. Hal ini dikarenakan tidak adanya sinar inframerah yang terpantul kembali ke sensor, sehingga lampu indikator menyala. Hasil dari pengujian ditampilkan melalui antarmuka pemrograman. Output menunjukkan bahwa sensor tidak berhasil mendeteksi objek pada jarak yang telah ditentukan, dengan keluaran pesan berulang yang menyatakan "Tidak ada Objek". Kondisi ini menunjukkan bahwa pada jarak tersebut, sensor berada di luar batas jangkauan deteksinya

4.1.4. Pengujian Motor Servo MG995

Pengujian motor servo bertujuan untuk mengevaluasi sejauh mana putaran sudut yang dihasilkan sesuai dengan tingkat akurasi yang tercantum dalam spesifikasinya. Dalam Tugas Akhir "Rancang Bangun Mouse Trap Berbasis Internet of Things (IoT) dengan Menggunakan Mikrokontroler ESP8266," pengujian ini akan mengukur besar perubahan sudut pada motor servo, yang berperan penting dalam mekanisme pengoperasian perangkat tikus, untuk memastikan bahwa setiap pergerakan dapat dilakukan dengan presisi yang dibutuhkan.

Tabel 4. 4 Pengujian Motor Servo 0°

Parameter	Nilai	Kondisi
Sudut Servo	0°	0°
Tegangan Servo	0.156V	Tegangan dalam posisi idle
Waktu respon	0.7 detik	Dari kondisi tertutup ke terbuka

Berdasarkan Tabel 4.4 hasil pengujian yang dilakukan, Hasil pengujian menunjukkan bahwa pada posisi 0° , tegangan yang terukur pada motor servo adalah sebesar 0.156V. Tegangan ini menunjukkan besarnya sinyal yang dikirim oleh mikrokontroler ESP8266 untuk mengatur dan mempertahankan posisi servo dalam kondisi terbuka. Tegangan yang rendah ini menggambarkan bahwa daya yang dibutuhkan oleh servo motor dalam posisi terbuka cukup kecil, sejalan dengan teori pengoperasian servo yang membutuhkan lebih sedikit daya ketika tidak ada beban gerakan.



Gambar 4. 5 Pengujian Motor Servo 0° menggunakan multimeter

Tabel 4. 5 Pengujian Motor Servo 90°

Parameter	Nilai	Kondisi
Sudut Servo	90°	90°
Tegangan Servo	0.322V	Tegangan untuk posisi tertutup
Waktu respon	0.5 detik	Dari kondisi terbuka ke tertutup

Berdasarkan Tabel 4.4 hasil pengujian yang dilakukan, Hasil pengujian menunjukkan bahwa pada posisi 90° , tegangan yang terukur pada motor servo adalah sebesar 0.322V, yang dimana besaran sinyal PWM yang dikirim oleh mikrokontroler. Nilai tegangan ini lebih tinggi dibandingkan dengan posisi 0° , menunjukkan bahwa servo memerlukan daya lebih besar untuk

mempertahankan posisi tertutup. Hal disebabkan oleh resistansi mekanis atau beban tambahan yang dialami servo saat pintu tertutup. Selain itu, waktu respon servo dari posisi terbuka (0°) ke posisi tertutup (90°) tercatat sekitar 0.5 detik.



Gambar 4. 6 Pengujian Motor Servo 90° menggunakan multimeter

4.2. Pengujian Sistem Keseluruhan

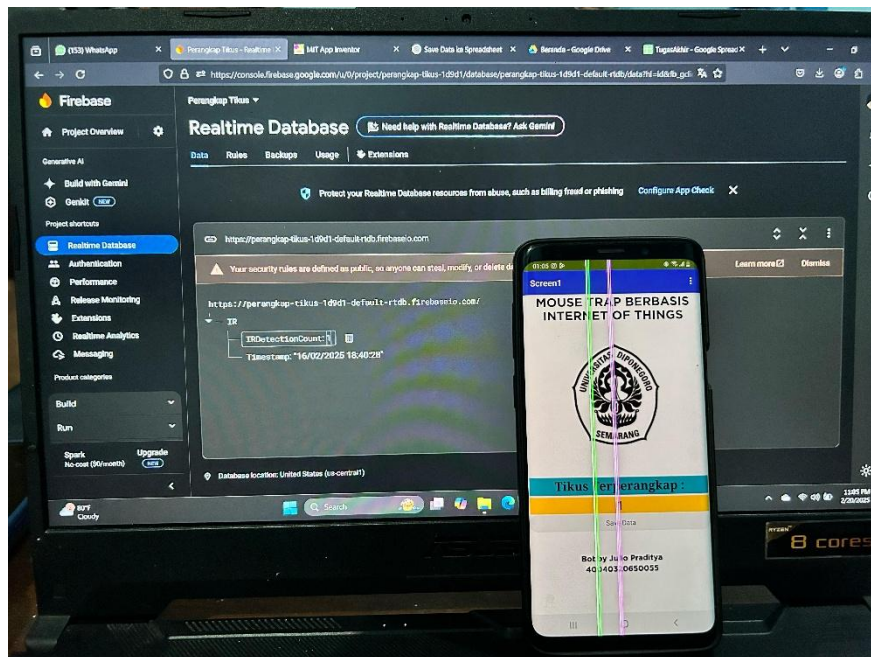
Pada pengujian keseluruhan ini bertujuan untuk memastikan bahwa alat perangkat tikus berbasis IoT yang telah dirancang dapat berfungsi dengan baik dan sesuai dengan tujuan pembuatan alat. Dalam proses pengujian ini dilakukan dengan menggunakan 3 tikus putih seperti yang ada pada gambar 4.7 sebagai media percobaannya dan jarak antara perangkat tikus dan router WiFi dalam pengujian ini adalah sekitar 10 meter. Tikus tersebut nantinya akan digunakan sebagai pengukuran dari alat jebakan tikus pada pengujian ini apakah alat ini sudah berjalan sesuai rencana. Selain itu, pengujian ini juga dilakukan terhadap sistem pengiriman notifikasi berupa pemantauan jumlah tikus yang berhasil terperangkap, yang ditampilkan melalui aplikasi yang dikembangkan menggunakan MIT App Inventor.



Gambar 4. 7 Tikus Putih sebagai media percobaan

4.2.1. Pengujian Counter

Pengujian counter pada alat perangkap tikus berbasis IoT dilakukan untuk memastikan fitur penghitung jumlah tikus yang tertangkap berfungsi dengan baik dan dapat dimonitor melalui aplikasi MIT App Inventor yang terintegrasi dengan Firebase. Counter ini bekerja dengan mendeteksi setiap aktivitas penangkapan tikus menggunakan sensor *IR Obstacle* yang terhubung dengan microcontroller ESP8266. Ketika tikus tertangkap, sensor akan mengirimkan sinyal ke microcontroller untuk menambahkan nilai pada counter, yang kemudian dikirimkan ke Firebase sebagai basis data. Firebase berperan sebagai media penyimpanan data secara *real time*, sehingga nilai counter yang tersimpan dapat diakses melalui aplikasi mobile. Berikut merupakan tampilan interface mit app inventor saat sensor ir obstacle membaca adanya objek.

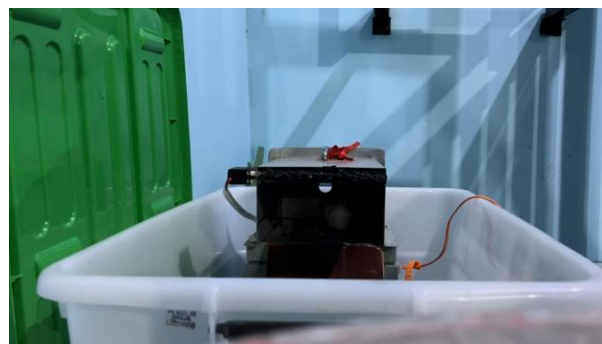


Gambar 4. 8 counter bernilai 1

Pada gambar 4.8 merupakan ketika sensor ir obstacle berhasil mendeteksi adanya objek, maka sensor akan mengirim data integer berupa nilai 1 ke firebase. Mit app inventor akan mengambil data dari firebase yang kemudian ditampilkan pada interface mit app inventor tersebut.

4.2.2. Pengujian alat dalam menangkap tikus

Pada Pengujian ini dilakukan secara realtime dengan 3 mencit atau tikus sebagai media percobaan. Pengujian ini dilakukan pada malam hari dari pukul 8 malam hingga pukul 9 malam.

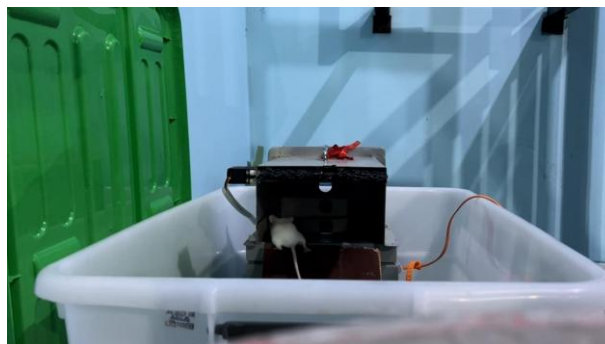


Gambar 4. 9 Tikus 1 terperangkap

Tabel 4. 6 Data logger tikus 1 terperangkap

IRDetectionCount	Timestamp
0	2025-03-11 20:00:03
0	2025-03-11 20:00:03
1	2025-03-11 20:11:51
1	2025-03-11 20:11:51

Pada Gambar 4.9, ditampilkan kondisi ketika tikus pertama berhasil masuk ke dalam perangkap. Data pada Tabel 4.6 menunjukkan bahwa pada pukul 20:00:03, nilai *IRDetectionCount* masih 0, yang menandakan bahwa sensor *infrared obstacle* (IR obstacle) belum mendeteksi adanya objek. Namun, pada pukul 20:11:51, nilai *IRDetectionCount* meningkat menjadi 1, yang menunjukkan adanya objek yang berhasil terdeteksi oleh sensor, dalam hal ini seekor tikus. Hal ini menegaskan bahwa sensor *infrared* berfungsi dengan baik dan mampu mengidentifikasi keberadaan tikus di dalam perangkap secara *real time*. Selain itu, sistem mencatat waktu deteksi dengan presisi, yang dapat digunakan untuk analisis lebih lanjut terkait aktivitas tikus di sekitar perangkap. Dengan demikian, data ini membuktikan bahwa sistem IoT berbasis mikrokontroler ESP8266 yang digunakan dalam perangkap ini bekerja sesuai dengan yang diharapkan dalam mendeteksi dan mencatat keberadaan tikus secara otomatis.

**Gambar 4. 10** Tikus 2 terperangkap

Tabel 4. 7 Data logger tikus 2 terperangkap

IRDetectionCount	Timestamp
1	2025-03-11 20:11:51
1	2025-03-11 20:11:51
2	2025-03-11 20:27:06
2	2025-03-11 20:27:06

Pada Gambar 4.10, ditunjukkan kondisi ketika tikus kedua berhasil masuk ke dalam perangkap setelah perangkap digunakan kembali. Berdasarkan data pada Tabel 4.7, pada pukul 20:11:51, nilai *IRDetectionCount* masih 1, menandakan bahwa satu tikus telah terdeteksi sebelumnya. Kemudian, pada pukul 20:27:06, nilai *IRDetectionCount* meningkat menjadi 2, yang menunjukkan bahwa tikus kedua telah berhasil masuk ke dalam perangkap. Saat tikus kedua melewati area sensor *infrared obstacle* (IR obstacle) yang terpasang di dalam perangkat, sensor kembali berfungsi dengan baik dan berhasil mendeteksi keberadaan objek baru di dalam perangkap. Sensor ini secara otomatis membaca pergerakan atau keberadaan objek yang masuk ke jalur deteksinya, yang dalam hal ini teridentifikasi sebagai seekor tikus tambahan. Setelah sensor mendeteksi objek, sinyal dikirimkan ke sistem untuk diproses lebih lanjut oleh mikrokontroler ESP8266. Keberhasilan sensor dalam membaca dan mengenali keberadaan tikus kedua ini menegaskan bahwa sistem pendeteksian bekerja secara konsisten dan akurat dalam mengidentifikasi objek yang masuk ke dalam perangkap. Selain itu, adanya perbedaan waktu antara deteksi pertama dan deteksi kedua menunjukkan bahwa perangkap tetap berfungsi setelah digunakan sebelumnya, memungkinkan sistem untuk menangkap lebih dari satu tikus dalam rentang waktu yang berbeda.



Gambar 4. 11 Tikus 3 terperangkap

Tabel 4. 8 Data logger tikus 3 terperangkap

IRDetectionCount	Timestamp
2	2025-03-11 20:27:06
2	2025-03-11 20:27:06
3	2025-03-11 20:43:29
3	2025-03-11 20:43:29

Pada Gambar 4.13, ditampilkan kondisi ketika tikus ketiga berhasil masuk ke dalam perangkap, menandakan bahwa perangkat masih berfungsi dengan baik dalam mendeteksi keberadaan tikus secara berulang. Berdasarkan data pada Tabel 4.8, pada pukul 20:27:06, nilai *IRDetectionCount* masih 2, menunjukkan bahwa dua tikus telah terdeteksi sebelumnya. Kemudian, pada pukul 20:43:29, nilai *IRDetectionCount* meningkat menjadi 3, yang berarti tikus ketiga telah berhasil masuk ke dalam perangkap. Saat tikus ketiga melewati area sensor yang telah dipasang di dalam perangkap, sensor *infrared obstacle* (IR *obstacle*) kembali beroperasi dan berhasil mendeteksi adanya objek baru di dalam ruang perangkap. Sensor ini bekerja dengan membaca perubahan pada jalur deteksinya akibat keberadaan suatu objek, dalam hal ini teridentifikasi sebagai seekor tikus tambahan. Setelah sensor mendeteksi keberadaan objek, sinyal secara otomatis dikirimkan ke mikrokontroler ESP8266 untuk diproses lebih lanjut. Informasi hasil dari sensor *infrared obstacle* ini kemudian dapat digunakan untuk memicu respons sistem, seperti

pencatatan jumlah tikus yang telah masuk ke dalam perangkap atau pengiriman notifikasi kepada pengguna melalui sistem berbasis *Internet of Things* (IoT).



Gambar 4. 12 Hasil 3 tikus terperangkap

Keberhasilan sensor dalam mendeteksi tikus ketiga ini menunjukkan bahwa sistem bekerja secara akurat dan konsisten, serta mampu mengidentifikasi objek yang masuk ke dalam perangkap tanpa mengalami kesalahan dalam pembacaan. Selain itu, perubahan nilai *IRDetectionCount* yang bertambah secara bertahap menegaskan bahwa perangkat dapat menangkap lebih dari satu tikus secara berurutan dalam rentang waktu yang berbeda, membuktikan efektivitas sistem dalam operasional jangka panjang.

4.2.3. Pengujian respon waktu counter pada mit app inventor

Pada pengujian ini merupakan waktu reaksi ketika data integer yang dikirim lalu ditampilkan pada mit app inventor. Tujuan dari pengujian ini adalah untuk mengetahui seberapa cepat data dapat dikirim dan diterima oleh aplikasi, serta untuk menilai kestabilan waktu pengiriman. Pengujian ini dilakukan menggunakan jaringan hotspot 4G atau wifi untuk mengirim data. Berikut analisis data diperlihatkan pada Tabel 4.9.

Tabel 4. 9 Pengujian respon waktu dalam pengiriman data integer ke mit app inventor

Percobaan Ke-	Waktu Delay Mengirim Data Integer Menggunakan Timer	
	<i>Counter</i>	Time
1	1	1 detik
2	2	0,5 detik
3	3	0,7 detik
Rata-rata		0,733 detik

Pada Tabel 4.9 disimpulkan bahwa Proses pengujian dilakukan sebanyak tiga kali. Dalam setiap percobaan, mikrokontroler mengirimkan data integer ke aplikasi dengan jeda waktu yang dikontrol menggunakan timer internal. Setiap waktu pengiriman dicatat untuk dianalisis.

$$t = \frac{1+0.5+0.7}{3} = \frac{2.2}{3} = 0,733 \text{ detik} \quad (4.1)$$

Hasil dari pengujian tersebut menunjukkan bahwa pada percobaan pertama, waktu *delay* yang dibutuhkan untuk mengirim data adalah 1 detik. Ini kemungkinan terjadi karena adanya proses inisialisasi awal komunikasi antara mikrokontroler dan aplikasi, atau sinkronisasi awal yang memerlukan waktu lebih lama.

Pada percobaan kedua, waktu *delay* menurun signifikan menjadi 0,5 detik. Hal ini menunjukkan bahwa sistem mulai berjalan lebih optimal setelah proses inisialisasi awal selesai. Sedangkan pada percobaan ketiga, waktu *delay* sedikit meningkat menjadi 0,7 detik, namun masih berada dalam batas toleransi dan menunjukkan performa yang cukup stabil. Dari ketiga percobaan tersebut, diperoleh rata-rata waktu delay sebesar 0,733 detik.

Waktu rata-rata ini menunjukkan bahwa sistem mampu mengirimkan data dengan respon waktu yang cukup cepat dan stabil. Beberapa faktor yang dapat memengaruhi nilai waktu *delay* ini antara lain adalah proses pemrosesan data oleh mikrokontroler, stabilitas koneksi jaringan Wi-Fi, responsivitas dari aplikasi MIT App Inventor itu sendiri.