Implementasi Subsistem Trap pada Camera Trap

Taufiq Abdullah Salim#1

#Program Studi Teknik Elektro, Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung  
Ganeca 10, Bandung, 40135, Jawa Barat, Indonesia

1Email: taufiqasalim@gmail.com

Abstrak**—** **Sistem *Camera Trap* adalah suatu system yang dapat mendeteksi adanya pergerakan makhluk hidup didepannya, dan kemudian mengambil gambar serta video dari makhluk hidup tersebut. Salah satu subsistem dari *Camera Trap* yaitu subsistem *trap* dan pengiriman trigger berfungsi untuk membaca pergerakan makhluk hidup yang terjadi di depannya lalu mengirimkan trigger pada subsistem kamera secara *wireless* yang kemudian akan diproses untuk mengambil data foto dan video.**

**Makalah ini berisi penjelasan mengenai rancangan, implementasi, dan pengujian subsistem *trap***

Kata kunci— *Camera Trap,* subsistem *trap, trigger*

1. **Pendahuluan**

Indonesia merupakan negara dengan luas area hutan mencapai 134 juta hektar atau sekitar 60 persen dari luas total Indonesia. Hutan yang ada di Indonesia memiliki peranan yang penting, tidak hanya sebagai sumber pembangunan ekonomi dan sumber kehidupan masyarakat saja, namun juga sebagai pemelihara lingkungan global.

Di Indonesia, kerusakan akibat faktor alam yang sering terjadi adalah kebakaran hutan. Kementerian Lingkungan Hidup, Departemen Kehutanan menyebutkan terdapat 561 titik api yang tersebar di 18 provinsi yang ada di Indonesia. Apabila titik api tersebut dapat dipantau secara secara kontinyu maka kasus kebakaran hutan dapat diminimalisir. Selain itu, faktor alam lainnya yang menjadikan kawasan hutan semakin kritis yang juga dapat mempengaruhi keberlangsungan hidup dari spesies flora dan fauna di dalamnya yaitu dapat berupa kelembapan, intensitas cahaya yang masuk, serta suhu hutan.

Faktor kedua yang menyebabkan kawasan hutan semakin berkurang adalah ulah manusia. Berdasarkan Data Kementerian Lingkungan Hidup, Departemen Kehutanan menyebutkan luas hutan Indonesia +133 juta hektar, namun kondisi saat ini 65% luas hutan tersebut sudah tidak berpohon atau gundul. Perburuan dan perdagangan ilegal satwa juga terus berlangsung untuk memenuhi permintaan pasar yang antara lain digunakan sebagai peliaharaan, dikonsumsi, dan untuk tujuan pengobatan tradisional. Menurut analisis WWF dan TRAFFIC (2003), nilai perdagangan tumbuhan dan satwa secara internasional (termasuk perdagangan ilegalnya) mencapai USD 159 miliar per tahun. Khusus untuk satwa yang dilindungi, nilai perdagangannya di singkat internasional mencapai US$20 miliar per tahun.

Dalam upaya menyelesaikan masalah kehutanan yang disebabkan faktor alam dan faktor manusia, saat ini digunakan polisi hutan yang bertugas untuk memantau, mengevaluasi, dan melakukan perlindungan terhadap hutan. Namun, dengan jutaan hektar hutan yang dimiliki Indonesia, hanya terdapat delapan ribu personel polisi hutan. Jumlah ini sangat terbatas, padahal mengacu pada data hasil studi Tigers Alive Initiative (TAI) diperlukan delapan orang polisi hutan setiap 100 km2. Kendala yang dihadapi dan dirasakan oleh polisi hutan saat ini yaitu petugas merasa menghadapi situasi yang membahayakan dikarenakan kurangnya perlengkapan untuk bekerja. Selain itu, polisi hutan merasakan bahwa dengan tingginya resiko yang dihadapi, imbalan yang didapatkannya belum memadai.

Indonesia memiliki banyak potensi dan masalah dalam menjaga hutan. Potensi yang ada di Indonesia berupa kepemilikan hutan yang sangat luas dengan keanekaragaman hayati yang sangat tinggi. Sedangkan masalah yang ada berupa masalah yang berasal dari faktor alam dan faktor manusia. Apabila hutan yang ada di Indonesia tidak dikonservasi dengan baik maka dapat menimbulkan kepunahan berbagai macam spesies. Oleh karenanya hutan seharusnya dikelola secara berkelanjutan agar dapat memberikan manfaat sebesar-besarnya bagi rakyat Indonesia, sebagaimana amanat Undang-Undang Dasar 1945.

Prasyarat menuju pengelolaan hutan secara lestari dan berkelanjutan, tidak terlepas dari kebutuhan data dan informasi yang lengkap, terpecaya, dan terkini. Informasi-informasi yang dibutuhkan berupa data kondisi lingkungan hutan seperti suhu, kelembapan, intensitas cahaya, dan lain-lain. Selain itu, juga diperlukan adanya informasi mengenai spesies flora dan fauna yang hidup di kawasan hutan. Pemantauan flora dan fauna ditujukan untuk menjaga spesies tersebut dari kepunahan akibat faktor alam maupun faktor manusia. Informasi-informasi yang dibutuhkan dalam mengelola hutan dapat didapatkan dengan cara pemasangan sistem pada hutan sehingga memungkinkan petugas untuk memantau melalui pusat konservasi. Sistem yang ada dapat berupa sistem kamera jebakan (camera trap) yang digunakan untuk mendapatkan data dan informasi untuk mengidentifikasi spesies tertentu.

1. **Tinjauan Pustaka**

Berikut beberapa studi literatur dalam perancangan dan implementasi subsistem *trap* pada sistem *Camera Trap*.

1. **NodeMCU**

*NodeMCU* merupakan mikrokontroller yang didalamnya sudah terintegrasi dengan modul WiFi yaitu ESP8266. Sehingga mikrokontoller ini dapat dengan mudah mengakses jaringan WiFi.

1. **Arduino IDE**

Arduino IDE merupakan software yang pada umumnya digunakan untuk melakukan pemrograman pada mikrokontroller Arduino. Software ini menggunakan Bahasa pemrograman Arduino yang berbasis C++. Board NodeMCU dapat ditambahkan pada Arduino IDE sehingga dapat dengan mudah melakukan pemrograman pada NodeMCU dari Arduino IDE.

1. **Raspberry Pi dengan Raspbian**

Raspberry Pi merupakan mikrokontroller multifungsi yang dapat digunakan untuk menjalankan program, bermain game, dan menjalankan beberapa aplikasi lain seperti halnya pada computer. Raspberry Pi yang digunakan akan diset dengan Operating System Raspbian. Semua pemrograman akan dibuat dan dijalankan dengan bantuan terminal yang terdapat pada Raspberry Pi

1. **Sensor Passive Infra-Red (PIR)**

Sensor PIR merupakan sensor pyroelektrik yang bekerja mendeteksi gerakan dengan mengukur perubahan sinar inframerah yang dihasilkan oleh objek didepannya. Sensor PIR memiliki material spesial yang sensitif terhadap cahaya inframerah. Sensor ini memiliki dua bagian utama yaitu IR sensor dan lensa fresnel.

IR sensor merupakan bagian yg berfungsi untuk mendeteksi adanya gerakan di area kerja sensor, sedangkan lensa Fresnel adalah lensa yang akan membuat tangkapan cahaya inframerah konvergen ke pusat sensor IR. Jangkauan dari sensor pun akan bisa lebih besar dengan adanya lensa tersebut. *Output* dari PIR sensor ini adalah PIR akan menghasilkan nilai HIGH untuk beberapa saat ketika membaca adanya pergerakan didepan sensor, selain kondisi tersebut, output dari PIR sensor akan bernilai LOW.

1. **Mosquitto MQTT**

MQTT atau Message Queue Telemetry Transport merupakan salah satu protocol berbasis *publish-subscribe.* Protokol ini bekerja diatas protocol TCP/IP. MQTT didesain untuk digunakan saat jaringan yang tersedia terbatas dan juga untuk pengiriman cepat dengan data berukuran relative kecil.

Mosquitto merupakan salah satu jenis MQTT yang dapat menerima dan mengirim pesan, Mosquitto adalah salah satu jenis MQTT yang didesain untuk bekerja paling ringan dan mudah.

1. **Spesifikasi, Perancangan, dan Implementasi**

Sistem *Camera Trap* terbagi menjadi dua bagian yaitu bagian *Camera Trap* dan bagian *Wireless Data Collection.*



Gambar 3. 1 Arsitektur *Camera Trap*

Pada bagian *Camera Trap* system akan terbagi lagi menjadi subsistem kamera dan subsistem trap. Akan terdapat 4 buah perangkat *trap* dan sebuah perangkat kamera pada setiap system *Camera Trap.* Pada subsistem trap, trap akan menunggu hingga terjadi gerakan oleh makhluk hidup, lalu mengirimkan *trigger* secara *wireless* kepada subsistem kamera. Pada subsistem kamera, *trigger* yang diterima akan diolah dan dilakukan pengambilan foto serta video dari objek yang dibaca tersebut.

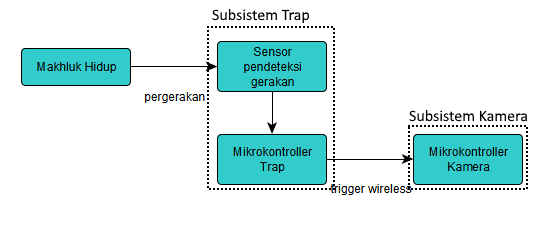
Foto dan video yang telah diambil akan disimpan pada memori subsistem kamera sebelum kemudian dilakukan pemindahan data secara *wireless* ke perangkat pembawa sebelum akhirnya akan dilakukan pemindahan data kembali dari perangkat pembawa tersebut ke computer pada data center dan kemudian akan dikirimkan secara otomatis pula dari computer di data center ke server *cloud*.

Dalam makalah ini, bahasan utama yang akan dijelaskan adalah pada subsistem trap dan pengiriman *trigger* dari trap tersebut ke subsistem kamera.

Spesifikasi dari perancangan susbsistem *trap* adalah sebagai berikut.

* Membaca gerakan makhluk hidup dan memberikan respon terhadap seluruh gerakan yang berjarak tidak lebih dari 5 meter didepan sensor
* Mengirimkan *trigger* ke subsistem kamera ketika membaca adanya gerakan untuk mengambil gambar atau video
* Mengambil foto maksimal 1 detik setelah sensor mendeteksi adanya gerakan

Berikut adalah ilustrasi *data flow diagram* subsistem *trap*.

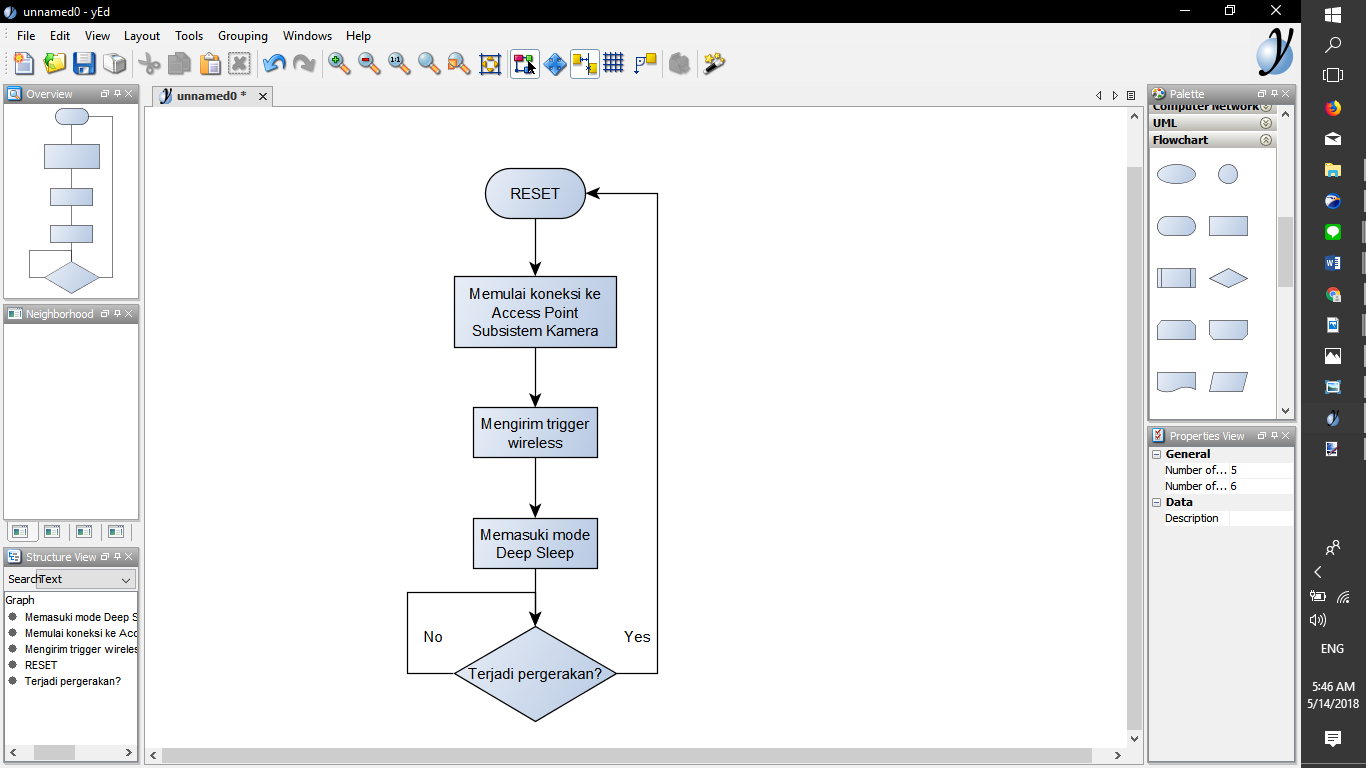
Gambar 3. 2 Data flow diagram

1. **Implementasi Subsistem Trap**

Proses perancangan pada subsistem *trap* ini menggunakan NodeMCU sebagai mikrokontroler dan HC-SR501 sebagai sensor *passive infra-red.* Pengaturan sensitivitas dan *delay* pada sensor diatur pada posisi seminimum mungkin namun dengan tetap memperhatikan agar sensitivitas sensor tidak terlalu sensitive hingga membaca gerakan yang sebenarnya tidak terjadi. *Delay* antardeteksi dari sensor PIR telah diukur kurang lebih 5,7 detik. Pengaturan jarak deteksi pada PIR ditentukan dengan mempertimbangkan area kerja kamera dan kondisi hutan yang terdapat banyak penghalang. Pengaturan *delay* minimum ditentukan dengan mempertimbangkan bahwa seluruh gerakan yang terjadi harus dapat terdeteksi, sehingga proses pengambilan data dapat segera dilakukan.

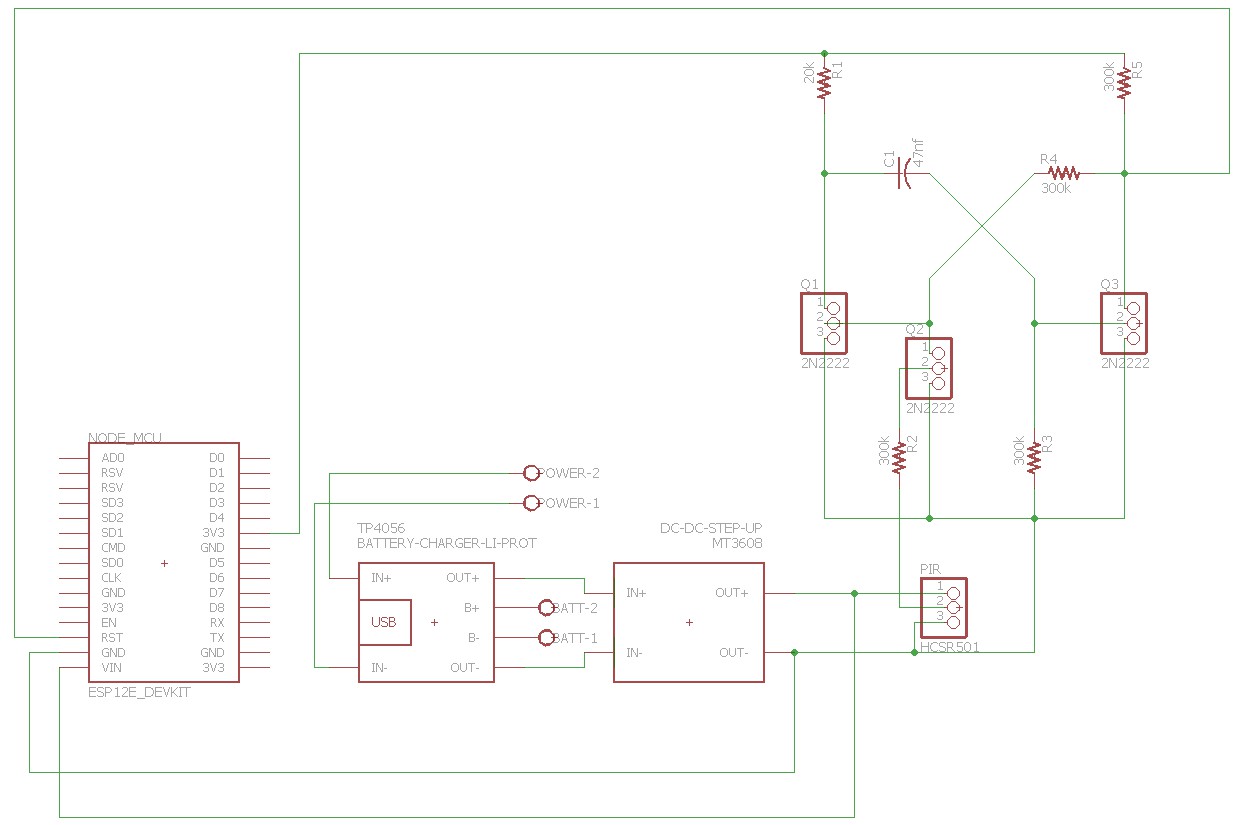
Selain itu pada perangkat keras sensor, dipilih mode normal pada pilihan *retriggering*. Ketika mendapatkan *trigger*, mode ini akan mengeluarkan output *high* lalu kembali *low.* Sehingga apabila ada gerakan kontinu yang terjadi, sensor tetap akan mengirimkan data *trigger* pada interval waktu tertentu. Berikut adalah tampilan osiloskop berupa sinyal hasil pembacaan data sensor PIR pada satu kali *trigger*.

Algoritma utama yang digunakan pada Subsistem *Trap* adalah sebagai berikut.



Gambar 3. 3 Algoritma Utama Trap

Algoritma Utama pada trap dimulai setiap kali terjadi reset pada mikrokontroller NodeMCU trap. Hal ini dilakukan dengan cara menambahkan rangkaian khusus diantara sensor PIR dengan pin reset dari NodeMCU yang akan menyebabkan tegangan keluaran rangkaian tersebut me-reset NodeMCU setiap kali nilai keluaran sensor PIR menjadi HIGH. Skematik dari rangkaian tersebut adalah sebagai berikut.



Gambar 3. 4 Skematik rangkaian reset

Kemudian setelah menerima reset dari sensor PIR NodeMCU akan mencoba untuk terhubung pada akses point yang telah dibuat oleh subsistem kamera yaitu akses point dengan nama CameraTrap.

Setelah berhasil terhubung, trap akan melakukan pengiriman *trigger* dengan cara mengirimkan perintah *publish* melalui Mosquitto MQTT ke server MQTT yang juga berada pada subsistem kamera.

Setelah trigger berhasil dikirim, NodeMCU akan memasuki mode deep-sleep, yaitu mode pada mikrokontroller NodeMCU dimana suplai daya yang digunakan sangatlah kecil.

Rangkaian reset akan berguna untuk kembali mereset NodeMCU ketika terdeteksi gerakan dan membuat algoritma terulang kembali dari atas.

1. **Implementasi Pengiriman Trigger dari Trap ke Kamera**

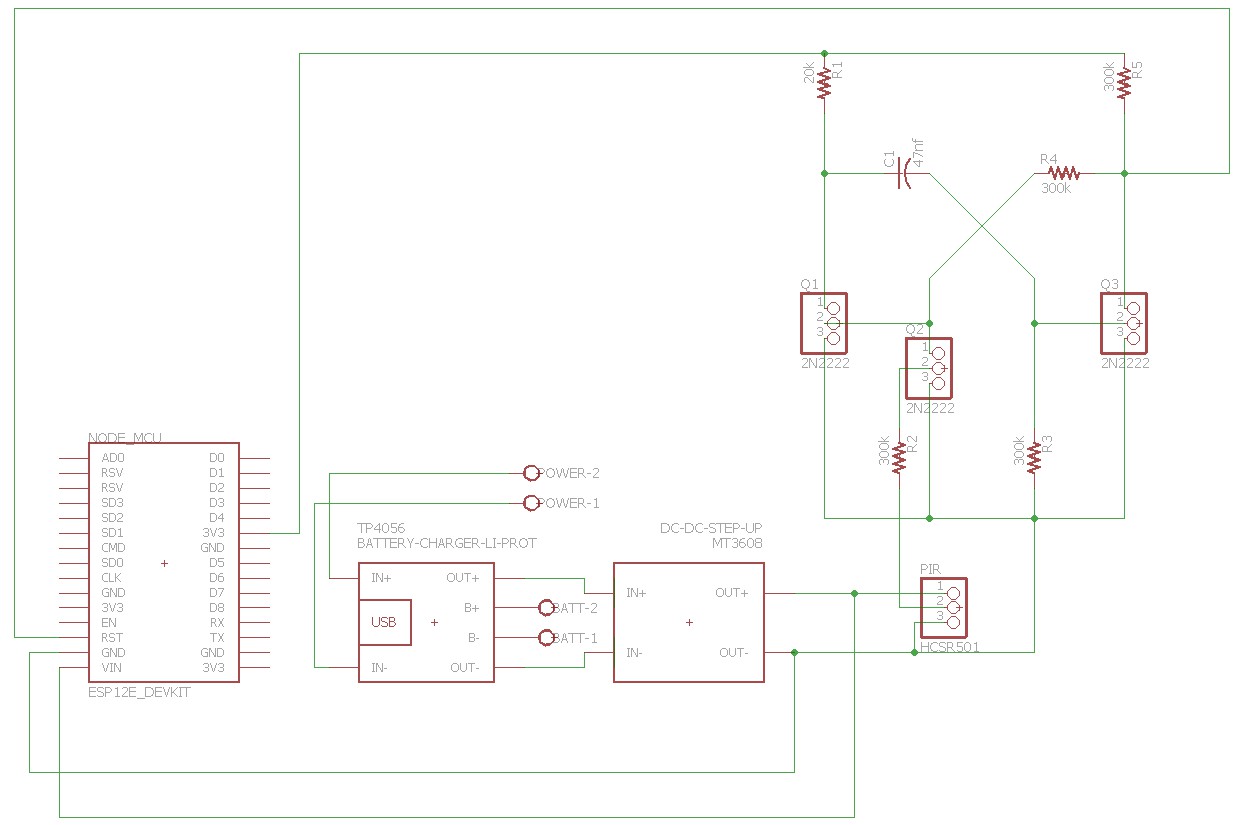
Implementasi pengiriman *trigger* dilakukan menggunakan protocol MQTT karena komunikasi yang diinginkan adalah komunkikasi yang cepat, ringan, dan tidak mengirim data banyak.

Implementasi ini terdari atas dua bagian, yaitu bagian *publish* dan bagian *subscribe.* Bagian *publish* merupakan bagian yang dilakukan oleh subsistem *trap* untuk mengirimkan sebuah data *string* kepada sebuah topik pada server MQTT yang kemudian akan dibaca oleh semua perangkat yang telah men-*subscribe* topik tersebut. Sementara bagian *subscribe* akan dilakukan oleh subsistem kamera untuk membaca data dari server MQTT setiap terdapat data baru pada topik yang telah di*subscribe* tersebut. Dalam hal ini, nama topik yang digunakan adalah topik “TrapNotif’.

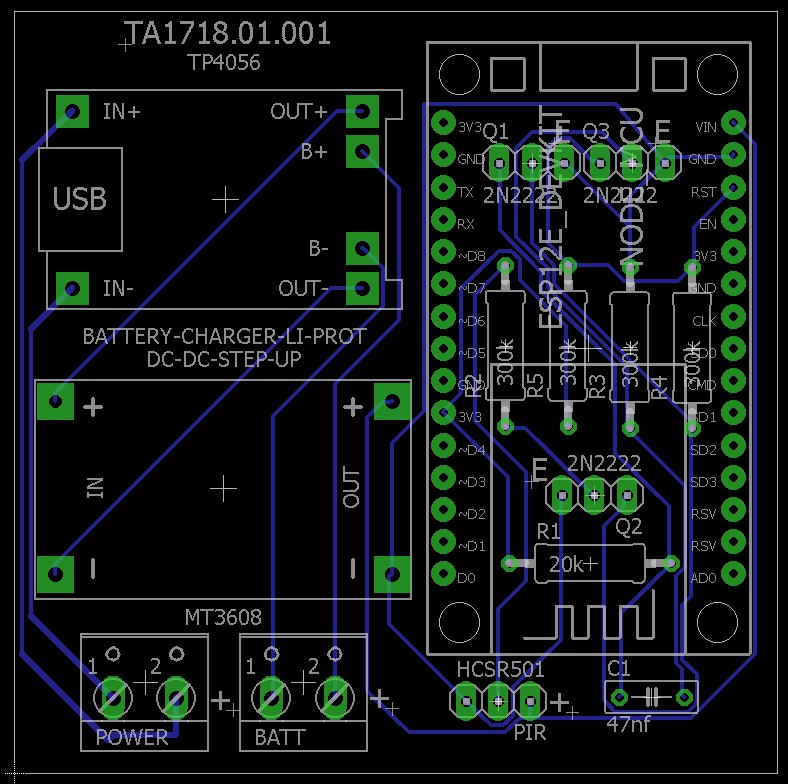
Algoritma *publish* pada NodeMCU terdapat dalam proses pengiriman *trigger wireless* pada Gambar 3.3. Sementara algoritma *subscribe* pada subsistem kamera akan membaca suatu *string* yang diterima pada topik TrapNotif. Kemudian string tersebut akan dibagi menjadi tipe notifikasi dan ID dari perangkat *trap* pengirim notifikasi. Setelah itu tipe notifikasi akan menjalankan program yang bersangkutan untuk menangani notifikasi tersebut. Sejauh ini notifikasi yang telah diimplementasi hanya satu, yaitu notifikasi *trigger* dimana jika mendapat notifikasi ini, subsistem kamera akan menjalankan program untuk melakukan pengambilan foto dan video.

1. **Implementasi Rangkaian dan Casing Trap**

Rangkaian yang digunakan untuk subsistem *trap* diimplementasikan dengan PCB dengan skematik dan *layout* PCB sebagai berikut.

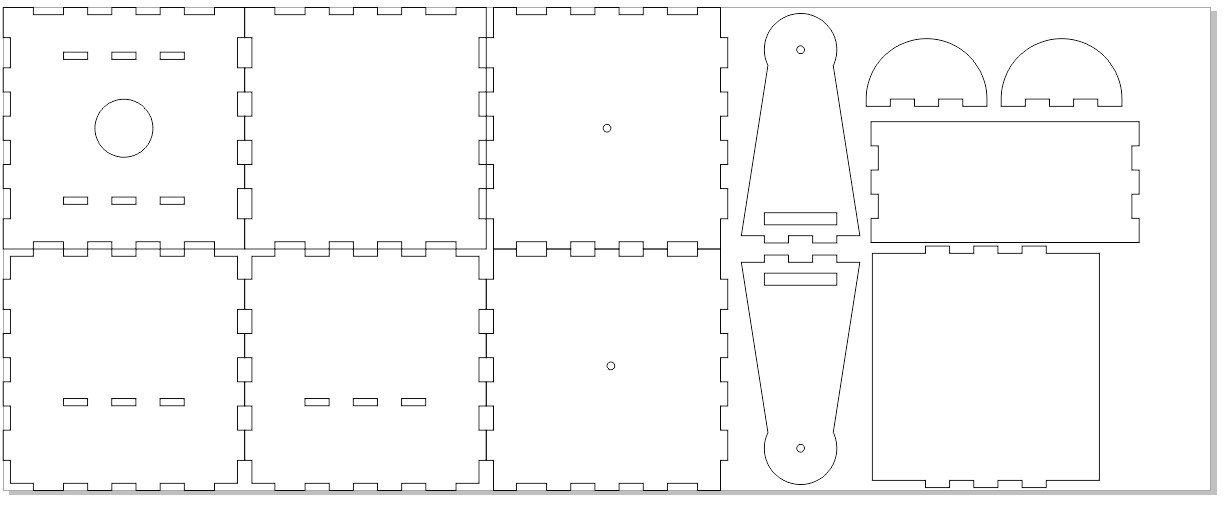


Gambar 3. 5 Skematik rangkaian *trap*



Gambar 3. 6 *Layout* rangkaian *trap*

Casing dari *trap* diimplementasikan dengan menggunakan *laser cut* pada bahan acrylic dan kemudian akan disatukan dengan menerapkan seal dengan bahan yang tidak mudah dimasuki air agar dapat menahan perubahan cuaca dan keadaan lain yang mungkin dialami *trap.* Desain dari casing trap pada *laser cut* adalah sebagai berikut.



Gambar 3. 6 *Layout* rangkaian *trap*

1. **Pengujian dan Verifikasi**

Pengujian pada subsistem *trap* dilakukan untuk menguji setiap spesifikasi yang harus dipenuhi. Konfigurasi pengujian setiap tahapnya yaitu sebagai berikut.

1. NodeMCU ESP8266 12-E core
2. Sensor *Passive Infra-Red* HC SR501
3. Roll Meter 10 meter dan 5 meter
4. Laptop Asus X455LD Intel Core i3-4030U 1.9GHz 6 GB RAM NVIDIA GeForce 820M 2GB VRAM
5. Power Bank Vivan Robot RT120 10000 mAh Dual USB
6. Baterai Lithium-ion model LGABE11865 dengan kapasitas 3200mAh dan tegangan nominal 3.75 V
7. iPhone 5S 16 GB Memory iOS 11.1.2
8. Serial Monitor pada Arduino IDE
9. Raspberry Pi Zero W (Raspberry Pi Zero W diatur sebagai Access Point untuk Raspberry Pi 3 Model B dan NodeMCU)
10. Software SSH Client (Putty & Shelly) serta VNC Viewer yang diatur untuk keperluan mengakses Raspberry Pi
11. Pengiriman trigger menggunakan Mosquitto MQTT dengan NodeMCU sebagai *publisher* dan Raspberry Pi Zero W sebagai *subscriber*

Prosedur yang dilakukan pada pengujian ini akan dilakukan pada dua kondisi lingkungan, yaitu kondisi lingkungan kampus (ada bangunan, pencahayaan utama dari lampu (±50 lux)) dan juga kondisi lingkungan hutan (dilakukan di Taman Hutan Raya Juanda Bandung, lingkungan dikelilingi oleh pohon, rumput dan tanaman lain, pencahayaan utama dari matahari (±90 lux)).

1. **Pengujian Jarak Deteksi Objek**

Pada pengujian ini akan dilihat keberhasilan sensor membaca seluruh gerakan yang terjadi di area kerja. Pada pengujian ini telah didapatkan nilai jarak maksimum yang dapat dideteksi (best case), nilai jarak minimum yang dapat dideteksi (worst case), serta nilai jarak rata-rata dari seluruh pengujian jarak deteksi trap. Diharapkan nilai jarak rata-rata dan jarak minimum cukup besar sehingga dapat menempatkan trap pada posisi optimal yang dapat membuat area total deteksi menjadi besar pula. Hasil yang didapatkan adalah sebagai berikut.

Tabel 4. 1 Hasil Pengujian di kawasan ITB

|  |  |  |  |
| --- | --- | --- | --- |
| Pengulangan ke- | Jarak (m) | Pengulangan ke- | Jarak (m) |
| 1 | 8.1 | **16** | 6.9 |
| 2 | 6.4 | **17** | 8.7 |
| 3 | 7.7 | **18** | 8.6 |
| 4 | 7.6 | **19** | 8.6 |
| 5 | 8.5 | **20** | 9.4 |
| 6 | 9.1 | **Ket:** | |
| 7 | 9.4 | **Data yang didapatkan hanya berjumlah 20 karena saat melakukan pengujian, banyak orang yang melewati dan melakukan aktivitas di lokasi yang digunakan sehingga pengambilan data menjadi terganggu** | |
| 8 | 9.1 |
| 9 | 11.4 |
| 10 | 7.9 |
| 11 | 10.7 |
| 12 | 9.5 |
| 13 | 8.4 |
| 14 | 10 |
| 15 | 7.5 |

Tabel 4. 2 Hasil Pengujian di Kawasan Tahura

|  |  |  |  |
| --- | --- | --- | --- |
| Pengulangan ke- | Jarak (m) | Pengulangan ke- | Jarak (m) |
| 1 | 8.2 | **16** | 8.9 |
| 2 | 8.5 | **17** | 10 |
| 3 | 8.6 | **18** | 9.4 |
| 4 | 9.3 | **19** | 9.7 |
| 5 | 8.2 | **20** | 9.3 |
| 6 | 10.9 | **21** | 9.8 |
| 7 | 8.3 | **22** | 9.8 |
| 8 | 9.7 | **23** | 10.7 |
| 9 | 9.4 | **24** | 10.2 |
| 10 | 9.5 | **25** | 9.9 |
| 11 | 9.8 | **26** | 10.1 |
| 12 | 9.7 | **27** | 9.9 |
| 13 | 9.1 | **28** | 10.6 |
| 14 | 9.8 | **29** | 10.3 |
| 15 | 9.2 | **30** | 9.8 |

Dari pengujian tersebut dapat diambil jarak maksimum (best case), jarak minimum (worst case), serta jarak rata-rata sebagai berikut:

Tabel 4. 3 Kesimpulan pengujian jarak *trap*

|  |  |  |
| --- | --- | --- |
|  | Kondisi lingkungan kampus (meter) | Kondisi lingkungan hutan (meter) |
| Jarak Minimum | 6.4 | 8.2 |
| Jarak Maksimum | 11.4 | 10.9 |
| Jarak Rata-rata | 8.675 | 9.551612903 |

Terlihat bahwa pengukuran jarak pada kondisi lingkungan hutan jauh lebih stabil dibandingkan dengan kondisi lingkungan kampus, hal tersebut kemungkinan dikarenakan banyaknya factor yang dapat mengganggu performa trap pada kondisi lingkungan kampus seperti pencahayaan lampu kurang baik dan tidak stabil, terdapat bangunan dan benda-benda lain yang dapat mengganggu hasil pembacaan sensor trap, dan lainnya. Namun, terlihat pula bahwa kedua kondisi, semua perhitungan jarak telah melebihi jarak minimum yang telah ditentukan pada spesifikasi yaitu 5 meter sehingga dapat dinyatakan performa trap telah memenuhi spesifikasi dengan sangat baik.

1. **Pengujian Pengiriman Trigger**

Pada pengujian diharapkan trap dapat dengan lancar mengirimkan trigger kepada camera. Serta didapat nilai delay cukup kecil sehingga dapat secepat mungkin mengambil foto dan video dari makhluk hidup yang menjadi objek deteksi.

Hasil yang didapatkan dari pengujian adalah trigger telah dapat dikirim dengan baik dari trap ke kamera dengan menggunakan MQTT pada topik “TrapNotif”. Selain itu, hasil pengujian *delay* pengiriman *trigger* adalah sebagai berikut.

Tabel 4. 4 *Delay* pengiriman *trigger*

|  |  |  |  |
| --- | --- | --- | --- |
| Pengulangan ke- | *Delay* (ms) | Pengulangan ke- | *Delay* (ms) |
| 1 | 347 | **16** | 1156 |
| 2 | 363 | **17** | 254 |
| 3 | 237 | **18** | 219 |
| 4 | 370 | **19** | 534 |
| 5 | 873 | **20** | 1371 |
| 6 | 1159 | **21** | 659 |
| 7 | 1338 | **22** | 329 |
| 8 | 656 | **23** | 554 |
| 9 | 328 | **24** | 575 |
| 10 | 230 | **25** | 741 |
| 11 | 462 | **26** | 455 |
| 12 | 483 | **27** | 365 |
| 13 | 773 | **28** | 264 |
| 14 | 1474 | **29** | 345 |
| 15 | 417 | **30** | 238 |

Dari pengujian tersebut dapat diambil delay maksimum (worst case), delay minimum (best case), serta delay rata-rata sebagai berikut:

Tabel 4. 5 Kesimpulan *delay* pengiriman *trigger*

|  |  |
| --- | --- |
| *Delay* Minimum | 219 |
| *Delay* Maksimum | 1474 |
| *Delay* Rata-rata | 598.9666667 |

Terlihat dari hasil pengujian delay bahwa delay dari pengiriman trigger tidak stabil, factor terbesar yang memengaruhi hal tersebut adalah delay antara NodeMCU direset hingga NodeMCU menemukan dan berhasil terhubung pada access point yang dibuat Raspberry Pi. Namun, terlihat pula bahwa delay rata-rata yang diperoleh cukup kecil yaitu sebesar 599 ms yang tidak terlalu besar memengaruhi performa sistem *Camera Trap* ini. Sehingga dapat dikatakan pengiriman trigger telah dapat dilakukan dengan cukup baik dan telah memenuhi spesifikasi.

1. **Simpulan**

Dari hasil pengujian yang dilakukan, subsistem *trap* telah berhasil membaca adanya pergerakan makhluk hidup didepannya dan mengirimkan *trigger* ke subsistem kamera. Subsistem ini dapat mendeteksi seluruh gerakan yang terjadi pada saat pengujian dan telah berhasil mengirimkan seluruh data ­*trigger* tersebut ke kamera dengan rata-rata durasi pengiriman adalah 600 millisecond. Selain itu pada gerakan yang kontinu, proses pembacaan gerakan memiliki toleransi waktu interval yang dibatasi antardeteksi oleh kemampuan sensor PIR yaitu sekitar 5.7 detik.

Referensi

1. Herlambang, Billie Naldo, dkk. B400-02-TA1617.01.070. ITB: Bandung
2. HC-SR501 PIR Motion Detector Datasheet
3. Ada, Lady. 2016. *PIR Motion Sensor – Pyroelectric (“Passive”) InfraRed Sensor.* <https://learn.adafruit.com/pir-passive-infrared-proximity-motion-sensor/overview>, diakses pada 13 Mei 2018.
4. Camacho, Luis, Design and Deployment of a Multimedia Wireless Sensor Network for Wildlife Inventory in Western Amazon Rain Forest. 2015.
5. NodeMCU Datasheet
6. Penjelasan MQTT . <https://en.wikipedia.org/wiki/MQTT>, diakses pada 13 Mei 2018
7. Penjelasan Mosquitto MQTT. <https://mosquitto.org/>. diakses pada 13 Mei 2018