

PERANCANGAN SISTEM DETEKSI API DENGAN *QUADCOPTER AR DRONE* MENGGUNAKAN METODE PENCARIAN *GRID*

Rishang Asmadi Sahita, Gembong Edhi Setyawan, S.T., M.T, D.Sc. Fatwa Ramdani, S.Si., M.Sc.
Program Studi Teknik Komputer
Fakultas Ilmu Komputer
Universitas Brawijaya
Malang, Indonesia
Rishang.asmadi@gmail.com

Abstrak – Api adalah salah satu unsur utama di alam kita dan merupakan pendukung kehidupan manusia. Dalam kehidupan sehari-hari, api sangat bermanfaat untuk memenuhi kebutuhan, seperti memasak dan menghangatkan diri. Banyak musibah yang terjadi apabila api tidak dapat dikendalikan, seperti kebakaran. Untuk mengatasi musibah yang ditimbulkan api, setiap negara memiliki personil pemadam kebakaran. Pemadaman api adalah misi yang berbahaya bagi pemadam kebakaran. Belakangan ini *autonomous* robot dipergunakan untuk memperkecil resiko dalam operasi pemadaman. Umumnya robot yang digunakan adalah *rover* pemadam api menggunakan sensor asap.

Sejak diperkenalkan *UAV (Unmanned Aerial Vehicle)*, yang merupakan salah satu jenis robot penjelajah udara tanpa awak, perkembangannya menjadi sangat pesat seiring dengan sistem *autopilot UAV* yang dapat diintegrasikan dengan sensor-sensor yang tersedia. Pada penelitian ini dilakukan pengembangan sistem *quadcopter* cerdas yang dapat mencari dan mendeteksi api dengan menggunakan sensor api. Pencarian api pada penelitian ini menggunakan metode pencarian *grid*. Sistem ini menggunakan sebuah sensor api yang dikoneksikan pada *Arduino Nano*. Data-data sensor akan dikirimkan secara serial ke *mainboard quadcopter* melalui *Arduino*. Pada *mainboard quadcopter* dipasang *API Node.JS* yang akan digunakan untuk membaca data sensor yang dikirimkan secara serial melalui *Arduino*, kemudian data-data tersebut dikonversikan menjadi *command* yang akan langsung dieksekusi oleh *mainboard quadcopter* untuk mendeteksi api. Sistem ini melakukan pencarian api dengan cara terbang *autonomous* dan melakukan pencarian menggunakan metode *grid* dengan waktu tempuh selama 7.4 detik sampai 8.2 detik untuk skenario pertama dengan posisi titik api terjauh, dan 3.0 detik sampai 5.3 detik untuk skenario kedua dengan posisi titik api pada jarak terdekat.

Kata Kunci – pendeteksi api, *quadcopter*, metode *grid*, sensor api, Node.JS

I. LATAR BELAKANG

Api adalah salah satu unsur utama di alam kita dan merupakan pendukung kehidupan manusia. Namun, banyak musibah yang terjadi apabila api tidak dapat dikendalikan. Untuk mengatasi akibat dari musibah yang disebabkan oleh tidak terkendalinya api, maka di tiap negara di dunia memiliki personil pemadam kebakaran. Pemadaman api adalah misi yang berbahaya bagi pemadam kebakaran. Setiap misi pemadaman beresiko kehilangan nyawa. Akhir-akhir ini *autonomous* robot dipergunakan untuk memperkecil resiko dalam operasi pemadaman. Umumnya robot yang digunakan adalah *rover* pemadam api menggunakan sensor asap.

Sejak diperkenalkan *UAV (Unmanned Aerial Vehicle)*, perkembangannya menjadi sangat pesat seiring dengan sistem *autopilot UAV* yang dapat diintegrasikan dengan sensor-sensor yang ada. Banyak model skenario yang dapat dilakukan oleh *UAV* dan sensor. Peran teknologi *UAV* dan sensor dapat membantu dalam operasi pemadaman api.

Pada penelitian ini menggunakan peran *UAV* dan sensor deteksi api untuk pendeteksian api. *UAV* yang paling stabil dan sangat fleksibel adalah *Quadcopter* (Austin, 2010). Pada penelitian ini dilakukan pengembangan sistem *quadcopter* cerdas yang dapat mencari dan mendeteksi api dengan menggunakan sensor api. *Quadcopter* yang digunakan dalam penelitian ini adalah *Parrot AR Drone 2.0*. Sistem ini menggunakan sebuah sensor api yang dikoneksikan pada *Arduino Uno*. Posisi sensor api dipasangkan menghadap kebawah untuk mengambil data dari api. Data-data sensor akan dikirimkan secara serial ke *mainboard quadcopter* melalui *Arduino*. Pada *mainboard quadcopter* dipasang *API Node.JS* yang akan digunakan untuk membaca data sensor yang dikirimkan secara serial melalui *Arduino*, kemudian data-data tersebut dikonversikan menjadi *command* yang akan langsung dieksekusi oleh *mainboard quadcopter* untuk mendeteksi api.

Pencarian api menggunakan metode pencarian. Terdapat beberapa macam metode pencarian yang umum digunakan seperti *Spiral*, *Zone*, *Parallel*, *Grid* dan

Anemotaxis (berdasarkan udara). Dan metode pencarian api yang digunakan dalam penelitian ini adalah metode Grid.

II. RUMUSAN MASALAH

Berdasarkan penjelasan pada latar belakang, maka dihasilkan beberapa rumusan masalah sebagai berikut :

1. Bagaimana mengembangkan sensor api pada *quadcopter Parrot AR Drone 2.0* untuk dapat mendeteksi api?
2. Bagaimana performa dan akurasi dari pembacaan dan pengiriman data ?
3. Bagaimana respon waktu *quadcopter* untuk mendeteksi api berdasarkan pembacaan sensor?
4. Bagaimana jarak ideal untuk pembacaan api dari *quadcopter* ke titik api?
5. Bagaimana pencarian api menggunakan *quadcopter*?

III. TUJUAN

Dari rumusan masalah yang telah ada, didapatkan tujuan sebagai berikut :

1. Penggunaan sensor api UVTRON untuk mendeteksi api.
2. Membangun sistem pendeteksi api yang terjangkau dengan menggunakan AR Drone 2.0.

IV. BATASAN MASALAH

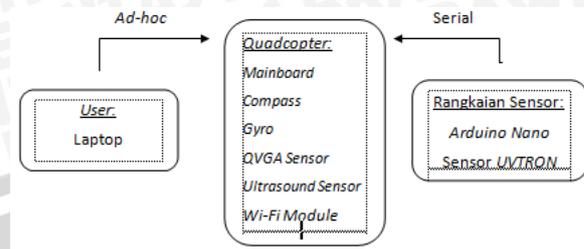
Agar penelitian dapat terfokus dan tidak menyimpang dari permasalahan, maka terdapat batasan-batasan masalah yang dibuat, yaitu :

1. Sensor yang digunakan adalah sensor UVTron R9454.
2. *Quadcopter* yang digunakan adalah AR Drone 2.0.
3. Pengujian menggunakan metode pencarian grid.
4. Sistem diuji dengan titik api dari lilin.
5. Ketinggian pengujian 1 sampai 2 meter dari titik api.
6. Pengujian dilakukan pada ruangan (Indoor).
7. Sistem hanya mendeteksi api tidak sampai memadamkan api..

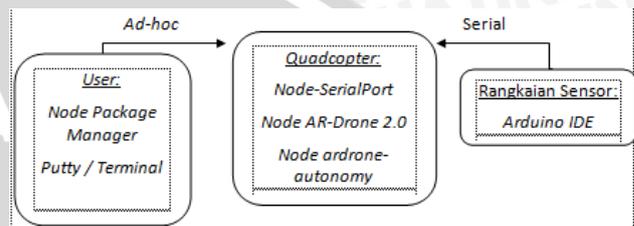
V. ANALISA KEBUTUHAN

Analisis kebutuhan merupakan tahap yang bertujuan untuk menganalisis semua kebutuhan yang diperlukan dalam membangun sistem yang akan diuji pada penelitian ini, sehingga penelitian dapat sesuai dengan apa yang diharapkan. Berikut ini merupakan beberapa kebutuhan sistem, yaitu

- Kebutuhan *hardware*



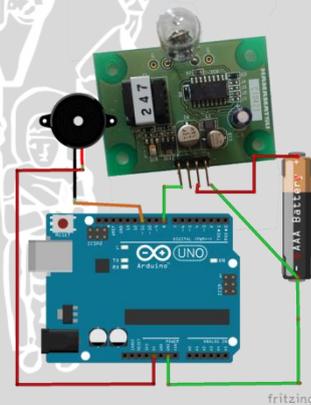
- Kebutuhan *software*



VI. PERANCANGAN SISTEM

Perancangan Perangkat Keras

Pada penelitian ini dilakukan perancangan perangkat keras yaitu perancangan sensor untuk sistem deteksi api. Adapun skematik dari rancangan rangkaian sensor dapat dilihat pada gambar 5.1.



Gambar VI.1 Skematik rancangan rangkaian sensor

Pada skematik dapat dilihat rangkaian sensor menggunakan *Arduino Uno* sebagai mikrokontroler dan *UVTRON R9454* sebagai sensor untuk mendeteksi api. Pin *ground UVTRON* dihubungkan ke pin *ground* pada *Arduino*, untuk *inputan* dari *wvtron* masuk ke pin 8 pada *Arduino*, dan untuk *output* untuk serial pada *quadcopter* dan sebagai *inputan* untuk buzzer terdapat pada pin 11 *Arduino*. Karena daya yang dibutuhkan sensor sebesar 12V maka menggunakan baterai 12V. Untuk

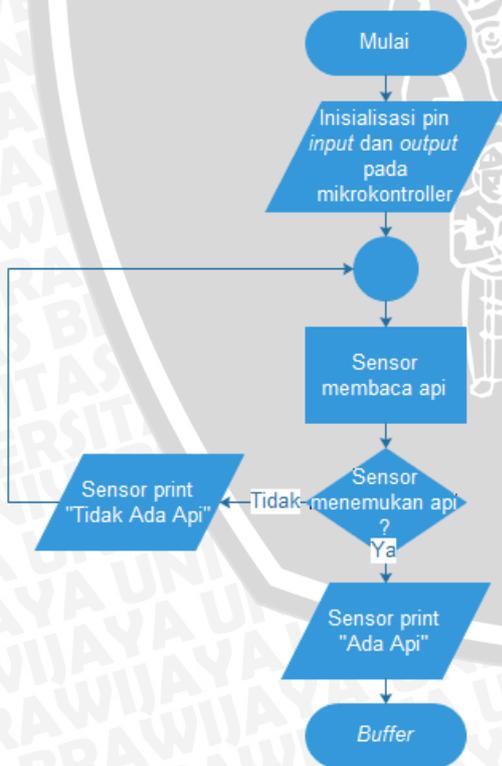
mikrokontroler menggunakan daya dari konektor *USB 5V* yang terdapat pada sistem *quadcopter*.

Perancangan Perangkat Lunak

Pada bagian perancangan perangkat lunak terdapat beberapa macam perancangan seperti perancangan perangkat lunak sistem deteksi api pada rangkaian sensor yang menggunakan *Arduino Uno*, selain itu perancangan perangkat lunak pada sistem *quadcopter* agar dapat menerima data pembacaan sensor dan mengubah data sensor menjadi *command* serta mengeksekusi *command* tersebut agar sistem dapat mencari dan mendeteksi api.

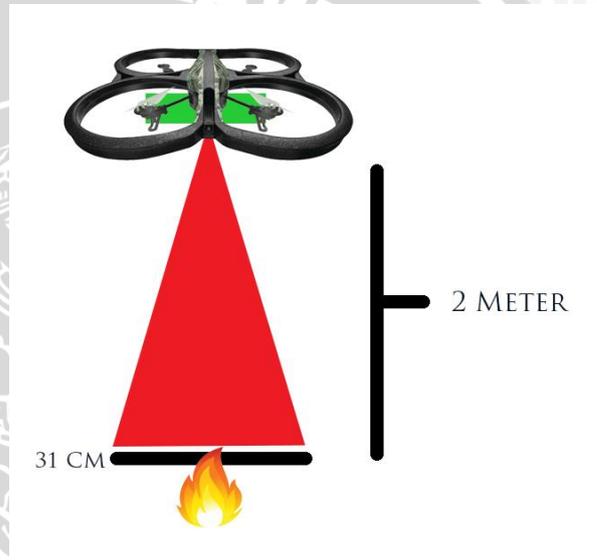
Perancangan Perangkat Lunak Pada Sistem Rangkaian Sensor

Agar sistem rangkaian sensor dapat bekerja sesuai dengan yang diinginkan, maka diperlukan perancangan perangkat lunak. Perancangan perangkat lunak pada sistem rangkaian sensor menggunakan *Arduino IDE*. Adapun penjelasan algoritma yang digunakan untuk mendeteksi objek pada sistem rangkaian sensor dapat dilihat pada gambar diagram alir 5.2.



Gambar VI.2 Diagram alir pada rangkaian sensor

Pada gambar diagram alir algoritma 5.2 dapat dilihat yang pertama kali dilakukan adalah penginisialisasian pin input dan output. Hal ini dilakukan agar sensor *UVTRON* dapat mengakuisisi dan mengirimkan data sesuai rancangan algoritma yang telah dirancang. Setelah inialisasi, maka sensor akan mulai membaca apakah ada sinar UV sesuai dengan ketinggian yang telah di rencanakan. Selanjutnya hasil pembacaan sensor di print menjadi tipe data *string*. Untuk pembacaan dengan hasil 1 atau “Ada Api” maka jumlah api harus sesuai dengan inputan *user* untuk selanjutnya *quadcopter* akan mendarat. Jika jumlah api tidak sesuai dengan inputan atau api tidak ditemukan, maka pembacaan api terus dilakukan. Rangkaian sensor akan terus mengirimkan hasil pembacaan deteksi api dan dikirimkan secara serial ke *mainboard AR Drone 2.0*.



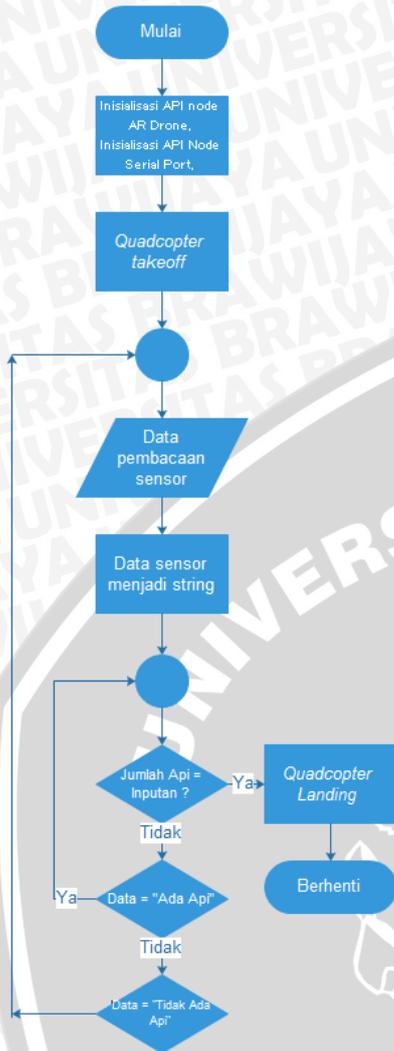
Gambar VI.3 Cakupan sistem dalam mendeteksi api

Pada gambar 5.3 dapat dilihat cakupan sistem dalam mendeteksi api membentuk kerucut dengan ketinggian terbang yang telah diatur setinggi 2 meter. *Quadcopter* akan berjalan secara perlahan dengan metode pencarian.

Perancangan Perangkat Lunak Pada Sistem *Quadcopter*

Pada *quadcopter* disematkan program yang telah dirancang menggunakan bahasa pemrograman *JavaScript*. Berikut diagram alir algoritma yang dirancang dapat dilihat pada gambar 5.4.





Gambar VI.4 Diagram alir algoritma pada sistem quadcopter

Dari gambar diagram alir 5.4 dapat dijelaskan yang pertama kali dilakukan algoritma yang dibangun dalam sistem quadcopter adalah penginisialisasian API Node AR Drone dan API Node Serial Port. Inisialisasi ini dilakukan agar fungsi yang ada pada Node AR Drone dapat dipanggil dan digunakan dalam kode program yang dibangun. Selanjutnya perintah *takeoff* diberikan kepada quadcopter. Pada saat terbang, sensor akan terus membaca adanya api atau tidak adanya api yang kemudian data tersebut diubah menjadi *string*. Jika data api sesuai dengan jumlah api yang diinputkan user maka quadcopter akan mendarat dan proses berhenti. Pengecekan jumlah api selalu dilakukan setiap sensor membaca api, jika tidak adanya api maka penelusuran dilakukan hingga menemukan api dan memenuhi syarat untuk berhenti.

Gambaran Kerja Sistem

Pada bagian ini menjelaskan bagaimana sistem yang sudah dirancang dapat bekerja. Adapun gambaran kerja sistem yang dijelaskan adalah bagaimana sistem mendeteksi objek yang didapatkan dari data rangkaian sensor kemudian bagaimana data sensor terbaca, dikonversi dan dieksekusi sehingga quadcopter dapat mendeteksi api. Gambaran kerja sistem dapat dilihat pada gambar 5.5.



Gambar VI.5 Diagram alir algoritma pada sistem quadcopter

Agar sistem dapat bekerja maka user harus terhubung dengan koneksi quadcopter secara wireless dan melakukan *telnet* ke 192.168.1.1. Setelah user terhubung dan masuk kedalam sistem operasi quadcopter, user dapat langsung menuliskan *command script* didalam path `./node`.

VII. IMPLEMENTASI SISTEM

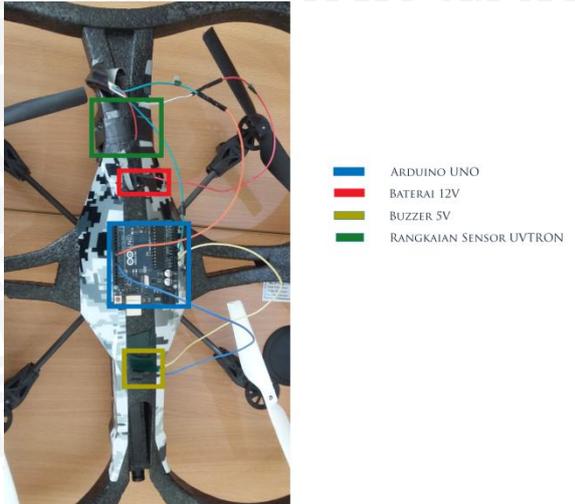
Tahapan implementasi adalah perealisasiian dari tahap perancangan sistem yang telah dilakukan sebelumnya. Implementasi sistem harus sesuai dengan perancangan untuk mendapatkan hasil yang sesuai dengan tujuan. Implementasi terbagi menjadi dua bagian yaitu implementasi *hardware* dan implementasi *software*.

Implementasi Hardware

Tahap implementasi *hardware* adalah tahap pembentukan sistem dari sisi perangkat keras. Implementasi *hardware* atau perangkat keras dimulai dengani pembuatan rangkaian sensor, prototype sistem, dan *hardware* sistem secara keseluruhan.

Rangkaian Sensor

Pada implementasi tahap ini dijelaskan tentang pemasangan sensor *UVTRON* dan juga buzzer gambar dari pengimplementasian rangkaian sensor dapat dilihat pada gambar 5.6.

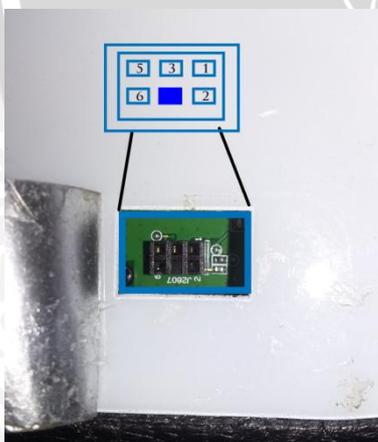


Gambar VII.1 Implementasi rangkaian sensor

Pada gambar 5.6 implementasi rangkaian sensor digunakan kabel jumper *male-female* untuk menghubungkan setiap komponen ke pin *arduino*.

Mainboard Quadcopter

Pada bagian implementasi *mainboard quadcopter* terdapat *pinout* yang berguna untuk menghubungkan antara rangkaian sensor yang telah dirancang dengan *quadcopter*. *Pinout* ini terdapat pada bagian bawah *quadcopter* yang ditutupi sticker aluminium.



Gambar VII.2 Pinout mainboard AR Drone 2.0

Pada gambar 5.7 dilihat bahwa *mainboard AR Drone 2.0* memiliki 6 buah *pinout* yang tiap pinnya berfungsi sebagai berikut:

1. LiPo 12V
2. VCC 5V
3. Ground
4. Tidak ada fungsi

5. TX (*Transmitter*)
6. RX (*Receiver*)

Pinout yang digunakan untuk rangkaian sensor adalah pin 6 atau pin RX (*receiver*) agar dapat berkomunikasi secara serial.

Implementasi Software

Tahap implementasi *software* adalah tahap pembentukan sistem dari sisi perangkat lunak. Implementasi *software* atau perangkat lunak dengan algoritma-algoritma yang sebelumnya telah dirancang.

Node.JS pada Sistem

Bagian ini akan menjelaskan tahapan untuk implementasi *Node.JS* pada sistem *quadcopter*. *API Node.JS* diletakkan pada direktori `/data/video` di sistem. Kemudian *permission Node.JS* diubah menggunakan `chmod +x` agar dapat dieksekusi. Pada gambar 5.8 dapat dilihat letak *Node.JS* pada sistem *quadcopter*.

```
BusyBox v1.14.0 () built-in shell (ash)
Enter 'help' for a list of built-in commands.

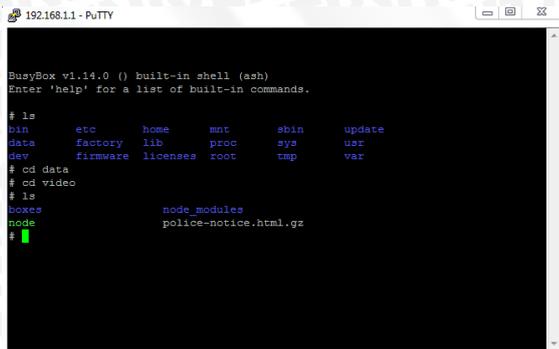
# cd data/video
# ./node
```

Gambar VII.3 Node.JS pada sistem quadcopter

API Node AR Drone 2.0 dan Node Serialport

Node AR Drone 2.0 dan *node Serialport* agar dapat difungsikan maka perlu dimasukkan kedalam sistem *quadcopter*. Karena *node AR Drone 2.0* dan *node serialport* merupakan sebuah *Node Package Manager* maka memerlukan direktori `node_modules` agar dapat dipanggil saat menjalankan program pada *Node.JS*. *Node_modules* memiliki lokasi yang sama dengan *Node.JS* yaitu pada direktori `/data/video`. Setelah direktori `node_modules` dibuat maka langkah selanjutnya adalah *copy API Node AR Drone 2.0* dan *node serialport* yang ada pada direktori *USB*. Direktori *USB* merupakan direktori dari *flashdrive* yang dipasangkan pada port *USB quadcopter*.

Untuk lebih jelasnya dapat dilihat pada gambar 5.9.



Gambar VII.4 Node.JS pada sistem quadcopter

Untuk hasil pembacaan yang lebih jelas dikumpulkan dalam tabel yang berisi jarak dan hasil pembacaan sensor pada tabel 6.1.

Tabel VIII.1 Tabel hasil pembacaan sensor

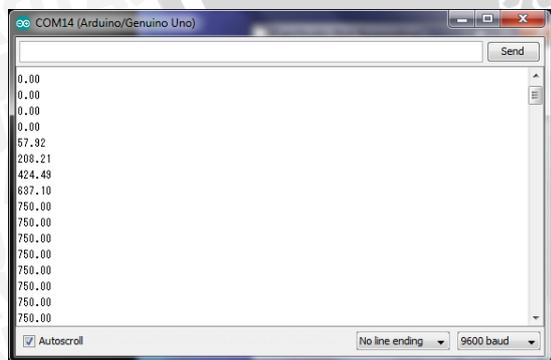
No.	Jarak Sensor Dengan Api (cm)	Pembacaan Sensor
1	0	750.00
2	50	750.00
3	100	750.00
4	150	750.00
5	200	750.00
6	250	722.87
7	300	547.00
8	350	286.60
9	400	140.76
10	450	82.11
11	500	25.66
12	550	2.20
13	600	0.00

VIII. PENGUJIAN

Pengujian Deteksi Api menggunakan Rangkaian Sensor

Pelaksanaan pengujian dilakukan dengan pengunggahan kode program deteksi api pada board arduino. Selanjutnya menghubungkan sensor UVTRON ke board arduino dan menghubungkan board arduino ke komputer menggunakan USB. Kemudian pada arduino IDE buka serial monitor untuk mengetahui hasil pembacaan dari sistem rangkaian sensor. Pengujian akan dilakukan untuk menemukan ketinggian yang optimal untuk mendeteksi titik api dan untuk menemukan lebar cakupan sensor.

Berdasarkan pengujian yang telah dilakukan, didapatkan data hasil pembacaan sensor menggunakan serial monitor arduino IDE sebagai berikut:

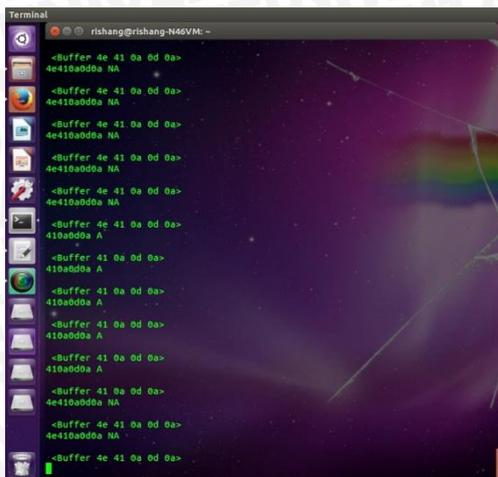


Gambar 6.1 Kode program pembacaan sensor untuk mendeteksi api

Pengujian Respon Waktu Penerimaan dan Pembacaan Data Pada Sistem Quadcopter Menggunakan API

Pelaksanaan pengujian ini menggunakan komputer user, rangkaian sensor, dan quadcopter. Komputer user harus terhubung kedalam jaringan AR Drone dan kemudian melakukan telnet ke alamat ip 192.168.1.1. Setelah masuk kedalam sistem quadcopter maka selanjutnya mengeksekusi Node.JS dengan perintah cd /data/video dan ./node untuk memasukkan kode program. Dari pengujian didapatkan hasil sebagai berikut:





Gambar VIII.2 Tampilan hasil dari pembacaan sensor pada sistem *quadcopter*

Pengujian *Quadcopter* Terbang Secara *Autonomous* Menggunakan *API Node Ar Drone*

Pelaksanaan pengujian ini menggunakan komputer *user* dan *quadcopter*. Komputer *user* harus terhubung kedalam jaringan *wireless Ar Drone* dan kemudian melakukan *telnet* ke alamat IP 192.168.1.1. Setelah terhubung kedalam jaringan *quadcopter*, *user* mengeksekusi *Node* dengan perintah `cd /data/video` dan `./node` untuk menuliskan program.

Hasil pengujian yang diperoleh adalah *quadcopter* dapat terbang dan melakukan pergerakan sesuai dengan perintah yang ada, walaupun terkadang tidak selalu sesuai pergerakannya dikarenakan *quadcopter Parrot Ar Drone 2.0* ini tidak dilengkapi dengan *gps* untuk kontrol posisi. Untuk kesesuaian pergerakan dengan perintah yang diberikan dapat dilihat pada tabel 6.1 untuk pergerakan *Forward*, tabel 6.2 untuk pergerakan *Backward*, tabel 6.3 untuk pergerakan *Right* dan tabel 6.4 untuk pergerakan *Left*.

Tabel VIII.2 Tabel hasil perintah *Forward* dengan kondisi aktual

Jarak Hasil Perintah (cm)	Jarak Aktual (cm)	Persentase Kesalahan
100 cm	100 cm	0%
95 cm	100 cm	5%
97 cm	100 cm	3%
100 cm	100 cm	0%
91 cm	100 cm	9%
100 cm	100 cm	0%

100 cm	100 cm	0%
Rata-rata persentase kesalahan		2.5%

Tabel VIII.3 Tabel hasil perintah *Backward* dengan kondisi aktual

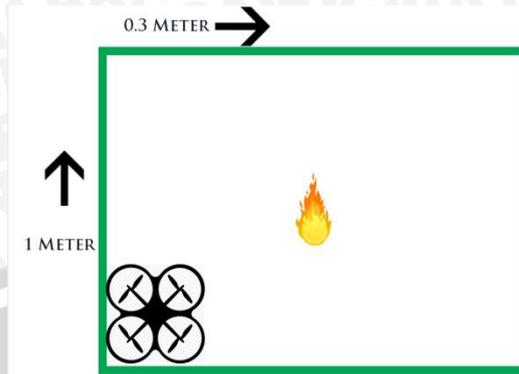
Jarak Hasil Perintah (cm)	Jarak Aktual (cm)	Persentase Kesalahan
90 cm	100 cm	10%
86 cm	100 cm	14%
100 cm	100 cm	0%
100 cm	100 cm	0%
99 cm	100 cm	1%
100 cm	100 cm	0%
100 cm	100 cm	0%
Rata-rata persentase kesalahan		3.5%

Tabel VIII.4 Tabel hasil perintah *Right* dengan kondisi aktual

Jarak Hasil Perintah (cm)	Jarak Aktual (cm)	Persentase Kesalahan
30 cm	30 cm	0%
25 cm	30 cm	17%
29 cm	30 cm	3.3%
30 cm	30 cm	0%
29 cm	30 cm	3.3%
30 cm	30 cm	0%
30 cm	30 cm	0%
Rata-rata persentase kesalahan		3.4%

Tabel VIII.5 Tabel hasil perintah *Left* dengan kondisi aktual

Jarak Hasil Perintah (cm)	Jarak Aktual (cm)	Persentase Kesalahan
30 cm	30 cm	0%
30 cm	30 cm	0%
30 cm	30 cm	0%
30 cm	30 cm	0%
30 cm	30 cm	0%
30 cm	30 cm	0%
28 cm	30 cm	6.6%
Rata-rata persentase kesalahan		0.9%



Hasil pengujian yang didapatkan dari pelaksanaan pengujian pencarian api dengan metode pencarian *grid* skenario 1 dihimpun pada tabel 6.6 dan untuk skenario 2 terdapat pada tabel 6.7. Tabel berisikan kumpulan waktu yang dibutuhkan *quadcopter* untuk mencapai titik api.

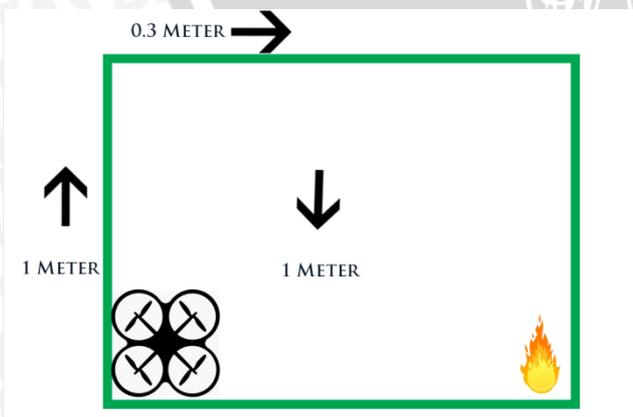
Tabel VIII.6 Tabel hasil respon waktu pencarian dan pembacaan api skenario 1

Percobaan ke-	Waktu yang Dibutuhkan (milisecond)
1	7700
2	8000
3	7400
4	7500
5	8200

Pengujian Pencarian Titik Api Menggunakan Metode Pencarian *Grid*

Pelaksanaan pengujian ini menggunakan komputer *user* dan *quadcopter*. Komputer *user* harus terhubung kedalam jaringan *wireless Ar Drone* dan kemudian melakukan *telnet* ke alamat IP 192.168.1.1. Setelah terhubung kedalam jaringan *quadcopter*, *user* mengeksekusi *Node* dengan perintah *cd /data/video* dan *.node* untuk menuliskan program.

Skenario Pengujian Pertama:



Skenario Pengujian Kedua:

Tabel VIII.7 Tabel hasil respon waktu pencarian dan pembacaan api skenario 2

Percobaan ke-	Waktu yang Dibutuhkan (milisecond)
1	4100
2	3802
3	3000
4	5300
5	4900

Berikut tampilan pada *terminal* komputer *user*.

```

Terminal
rishang@rishang-N4cVM: ~
T10ak: APT
var autonomy = require('ardrone-autonomy');
var mission = autonomy.createMission();
var APT;
var serialport = require('node-serialport')
var sp = new serialport.SerialPort('/dev/ttyO3', {
  parser: serialport.parsers.readline("\n"),
  baud: 9600
});
sp.on('data', function(chunk) {
  APT = chunk.toString();
  console.log("%s", APT);
});
mission.takeoff();
mission.altitude(1);
mission.hover(1000);
mission.forward(1);
mission.hover(1000);
mission.right(0.3);
mission.hover(1000);
mission.backward(1);
mission.hover(1000);

```

Gambar VIII.3 Tampilan terminal saat quadcopter menemukan api

IX. KESIMPULAN

1. Sistem ini terdiri dari dua komponen utama yaitu *quadcopter* dan rangkaian sensor. *Quadcopter* bertugas sebagai pelaksana misi pencarian api dan rangkaian sensor bertugas membantu *quadcopter* dalam pembacaan sinar UV yang dihasilkan oleh api. Hasil pembacaan sensor akan dikirimkan ke sistem *quadcopter* yang selanjutnya diubah menjadi perintah terbang.
2. Pembacaan api pada sensor menggunakan *Arduino IDE* dan sistem *quadcopter* membutuhkan waktu sebesar 100ms untuk membaca adanya api.
3. *AR Drone* merupakan *quadcopter* yang terjangkau, dengan *client protocol* bersifat *open source* dan dapat terhubung secara *wireless* dengan komputer *user*. Dengan menggunakan *Node.JS*, *user* dapat memprogram *quadcopter* agar terbang secara *autonomous*. Performa dan akurasi dalam pembacaan dan pengiriman data melalui serial terkadang mengalami *memory leak* yang menyebabkan sistem memutuskan konektivitas terhadap komputer *user*.
4. Berdasarkan pengujian dan penyesuaian didapatkan jarak ideal untuk pembacaan api dari sistem *quadcopter* menuju titik api sebesar 1 meter sampai dengan 2 meter.
5. Pencarian api dengan *quadcopter* menggunakan metode *grid* memiliki waktu tempuh selama 7.4 detik sampai 8.2 detik untuk mencapai titik api pada skenario pertama. Untuk skenario kedua, *quadcopter* membutuhkan waktu selama 3.0 detik sampai 5.3 detik.

X. DAFTAR PUSTAKA

- Austin Reg., 2010. *Unmanned Aircraft System: UAVS Design, Development and Deployment*. Wiley, UK.
- Barrow, Erik., 2014. *Autonomous Navigation and Search in an Indoor Environment Using an AR Drone*. [e-journal] (2014). Tersedia di: <https://www.academia.edu/9737574/Autonomous_Navigation_and_Search_in_an_Indoor_Environment_Using_an_AR_Drone> [Diakses pada 6 Juni 2016].
- Chang, Henry., Htut, Phyo., Ho, Michael dan Bandeira, Roberto., 2015. *Autonomous Quadcopter for Fire Detection*. [e-journal] (2015). Tersedia di:<www.robertobandeira.com/uploads/6/5/6/4/65641297/_fire_detection_on_drone.pdf> [Diakses pada 30 Mei 2016].
- C. Yuan, Y. Zhang and Z. Liu., 2015. *A survey on technologies for automatic forest fire monitoring, detection, and fighting using unmanned aerial vehicles and remote sensing techniques*.
- Eschenauer, Laurent., 2013. *Node ardrone-autonomy*. [API]. Tersedia di: <<https://github.com/eschnou/ardrone-autonomy>> [diakses pada 6 Mei 2016].
- Geisendorfer, Felix., 2013. *Node ar-drone*. [API]. Tersedia di: <<https://github.com/felixge/node-ar-drone>> [diakses pada 1 April 2016].