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

SKRIPSI

KEMINATAN TEKNIK KOMPUTER

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Rishang Asmadi Sahita
NIM: 125150301111008

PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2016

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

PENGESAHAN

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

SKRIPSI

KEMINATAN TEKNIK KOMPUTER

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Rishang Asmadi Sahita
NIM: 125150301111008

Skripsi ini telah diuji dan dinyatakan lulus pada
19 Agustus 2016
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I	Dosen Pembimbing II
Gembong Edhi Setyawan, S.T, M.T	D.Sc. Fatwa Ramdani, S.Si., M.Sc.
NIK: 201208 761201 1 001	NIK: -

Mengetahui
Ketua Jurusan Teknik Informatika

Tri Astoto Kurniawan, S.T, M.T, Ph.D
NIP: 197105182003121001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 2 Agustus 2016

Rishang Asmadi Sahita

NIM: 125150301111008

KATA PENGANTAR

Puji syukur kehadirat Allah SWT yang telah melimpahkan rahmat, taufik dan hidayah-Nya sehingga laporan skripsi yang berjudul "Perancangan Sistem Deteksi Api dengan *Quadcopter AR Drone* Menggunakan Metode Pencarian Grid" ini dapat diselesaikan dengan baik dan tepat waktu.

Penulis menyadari bahwa skripsi ini tidak akan berhasil tanpa bantuan dari beberapa pihak. Oleh karena itu, penulis ingin menyampaikan rasa terimakasih dan hormat kepada:

1. Bapak dan Ibu serta saudara-saudara tercintaku yang senantiasa memberikan nasehat, doa dan restunya serta semangat kepada penulis demi kesuksesan dan kelancaran studi penulis.
2. Bapak Gembong Edhi Setyawan, S.T, M.T dan Bapak D.Sc. Fatwa Ramdani, S.Si., M.Sc., selaku dosen pembimbing skripsi yang telah dengan sabar membimbing, mengarahkan dan memberikan semangat kepada penulis agar hasil skripsi ini bermanfaat untuk kedepannya.
3. Bapak Sabriansyah Rizqika Akbar, S.T, M.Eng selaku Ketua Program Studi Teknik Komputer Universitas Brawijaya Malang yang selalu memberikan semangat serta meng-*invite party* Dota2.
4. Seluruh dosen dan Civitas Akdemik yang berada di Laboratorium Sistem Komputer dan Robotika Fakultas Ilmu Komputer Universitas Brawijaya serta teman civitas akademika Fakultas Ilmu Komputer Universitas Brawijaya yang telah memberikan batuan.
5. Venezia Indra Ghassani yang selalu memberikan semangat dan menemani penulis selama menempuh pendidikan di Fakultas Ilmu Komputer Universitas Brawijaya.
6. Teman-teman kos Dinoyo Permai Kavling 7 yang turut membantu penulis dalam pengerjaan penelitian ini.
7. Seluruh teman-teman Sistem Komputer angkatan 2012 #SatuDulur, TekkoTech, LaL dan *Golden Emissary* Dota2 atas semangat serta segala bantuan yang telah diberikan.

Malang, 2 Agustus 2016

Penulis

Rishang Asmadi Sahita

ABSTRAK

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 kebarakan. 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

repository.ub.ac.id

ABSTRACT

Abstract *– Fire is one of the main elements in our nature and a human's life support. In a whole day, fire is very useful to meet a human needs, such as cooking and warming themselves from the cold. Numerous accidents happened caused by uncontrolled fire, such as wildfire. To overcome this problem, all country in this world have fire brigade. The process of fire extinguishing is a dangerous mission. Lately, an autonomous robot is used to minimize the risk in the extinguishing operation. Commonly used rover fire extinguisher robot which is using the smoke sensor.*

Since introducing UAV (Unmanned Aerial Vehicle), which is one of unmanned aerial robot explorers type, it grew very rapidly along with autopilot system. UAV can be integrated with many sensors. In this research, the development of intelligent quadcopter system that can search and detect fires using fire sensors. This system using a fire sensor that connected with Arduino Nano. Sensor data will be transmitted via serial to quadcopter's mainboard using Arduino. On the quadcopter's mainboard has installed Node.JS API that will be used to read the sensor data and send the data through serial using Arduino, then the data will be converted into flight command and will be executed by quadcopter mainboard to detect the fire. Fire search method which is used in this research is Grid search method. The system is able to search and detect fire by autonomous flight using grid pattern search with travel time 7.4 sec to 8.2 sec for first scenario farthest fire position and 3.0 sec to 5.3 sec for second scenario nearest fire position.

Keyword – *fire detection, quadcopter, grid method, fire sensor, Node.JS*

UNIVERSITAS BRAWIJAYA

DAFTAR ISI

PENGESAHAN ii
PERNYATAAN ORISINALITAS iii
KATA PENGANTAR iv
ABSTRAK v
ABSTRACT vi
DAFTAR ISI vii
DAFTAR TABEL x
DAFTAR GAMBAR xi
BAB 1 PENDAHULUAN 1
1.1 Latar Belakang 1
1.2 Rumusan Masalah 2
1.3 Tujuan 2
1.4 Manfaat 2
1.5 Batasan Masalah 2
1.6 Sistematika Pembahasan 2
1.7 Jadwal Penelitian 3
BAB 2 LANDASAN KEPUSTAKAAN 4
2.1 Tinjauan Pustaka 4
2.2 Dasar Teori 7
2.2.1 *Unmanned Aerial Vehicle* (UAV) *Quadcopter* 7
2.2.2 *Arduino Nano* 8
2.2.3 *Parrot AR Drone 2.0* 9
2.2.4 Sensor Api UVTRON R9454 11
2.2.5 *Application Programming Interface (API)* 12
BAB 3 METODOLOGI PENELITIAN 13
3.1 Metode Penelitian 13
3.1.1 Studi Literatur 14
3.1.2 Rekayasa Persyaratan 14
3.1.3 Perancangan Sistem 17
3.1.4 Implementasi Sistem 19

3.1.5 Pengujian dan Analisis 19
3.1.6 Penarikan Kesimpulan 19
BAB 4 REKAYASA PERSYARATAN 20
4.1 Pendahuluan 20
4.1.1 Tujuan 20
4.1.2 Ruang Lingkup 20
4.1.3 Istilah 20
4.1.4 Sistematika 21
4.2 Deskripsi Umum 21
4.2.1 Perspektif Sistem 21
4.2.2 Kegunaan 21
4.2.3 Karakteristik Pengguna 22
4.2.4 Lingkungan Operasi 22
4.2.5 Batasan Perancangan dan Implementasi 22
4.2.6 Asumsi dan Ketergantungan 22
4.3 Kebutuhan Antar muka Eksternal 22
4.3.1 Antarmuka Pengguna 22
4.3.2 Antarmuka Perangkat Keras 22
4.3.3 Antarmuka Perangkat Lunak 23
4.3.4 Kebutuhan Antarmuka Komunikasi 23
4.4 Kebutuhan Fungsional 23
4.5 Kebutuhan Non Fungsional 24
BAB 5 PERANCANGAN DAN IMPLEMENTASI 25
5.1 Perancangan Sistem 25
5.1.1 Perancangan Perangkat Keras 25
5.1.2 Perancangan Perangkat Lunak 26
5.2 Gambaran Kerja Sistem 29
5.3 Implementasi Sistem 29
5.3.1 Implementasi *Hardware* 29
5.3.2 Implementasi *Software* 33
BAB 6 PENGUJIAN 35
6.1 Pengujian Deteksi Api menggunakan Rangkaian Sensor 35

6.1.1 Tujuan Pengujian 35
6.1.2 Prosedur Pengujian 35
6.1.3 Pelaksanaan Pengujian 35
6.1.4 Hasil Pengujian 36
6.1.5 Analisis Pengujian 37
6.2 Pengujian Respon Waktu Penerimaan dan Pembacaan Data Pada Sistem *Quadcopter* Menggunakan *API* 37
6.2.1 Tujuan Pengujian 37
6.2.2 Prosedur Pengujian 37
6.2.3 Pelaksanaan Pengujian 38
6.2.4 Hasil Pengujian 39
6.2.5 Analisis Pengujian 39
6.3 Pengujian *Quadcopter* Terbang Secara *Autonomous* Menggunakan *API Node Ar Drone* 40
6.3.1 Tujuan Pengujian 40
6.3.2 Prosedur Pengujian 40
6.3.3 Pelaksanaan Pengujian 40
6.3.4 Hasil Pengujian 41
6.3.5 Analisis Pengujian 43
6.4 Pengujian Pencarian Titik Api Menggunakan Metode Pencarian *Grid* 43
6.4.1 Tujuan Pengujian 43
6.4.2 Prosedur Pengujian 44
6.4.3 Pelaksanaan Pengujian 45
6.4.4 Hasil Pengujian 46
6.4.5 Analisis Pengujian 47
BAB 7 PENUTUP 48
7.1 Kesimpulan 48
7.2 Saran 48
DAFTAR PUSTAKA 49

DAFTAR TABEL

Tabel 1.1 Jadwal Penelitian 3
Tabel 2.1 Tabel spesifikasi *Arduino Nano* 9
Tabel 2.2 Tabel spesifikasi *mainboard AR Drone 2.0* 10
Tabel 2.3 Tabel spesifikasi motor *AR Drone 2.0* 10
Tabel 2.4 Tabel spesifikasi sensor *UVTRON R9454* 11
Tabel 4.1 Tabel Istilah 20
Tabel 5.1 Tabel hasil pembacaan sensor 31
Tabel 6.1 Tabel hasil perintah *Forward* dengan kondisi aktual 42
Tabel 6.2 Tabel hasil perintah *Backward* dengan kondisi aktual 42
Tabel 6.3 Tabel hasil perintah *Right* dengan kondisi aktual 43
Tabel 6.4 Tabel hasil perintah *Left* dengan kondisi aktual 43
Tabel 6.5 Tabel hasil respon waktu pencarian dan pembacaan api skenario 1 ... 46
Tabel 6.6 Tabel hasil respon waktu pencarian dan pembacaan api skenario 2 ... 47

DAFTAR GAMBAR

Gambar 2.1 Kondisi ketinggian 4
Gambar 2.2 Konversi data pulsa ke desimal 5
Gambar 2.3 Perbandingan Metode Pencarian 6
Gambar 2.4 Metode pencarian *Grid* 6
Gambar 2.5 Rangka *Quadcopter* 7
Gambar 2.6 *Yaw, pitch dan roll* 8
Gambar 2.7 Ilustrasi pergerakan Rotor 8
Gambar 2.8 *Arduino Nano* 9
Gambar 2.9 *Parrot AR Drone 2.0* 11
Gambar 2.10 Sensor api UVTRON R9454 11
Gambar 3.1 Diagram alir penelitian 13
Gambar 3.2 Kebutuhan *Hardware* 15
Gambar 3.3 Kebutuhan *Software* 16
Gambar 3.4 Diagram blok sistem 17
Gambar 3.5 *Sequence Diagram* Sistem 18
Gambar 5.1 Skematik rancangan rangkaian sensor 25
Gambar 5.2 Diagram alir pada rangkaian sensor 26
Gambar 5.3 Diagram alir algoritma pada sistem *quadcopter* 28
Gambar 5.4 Diagram blok sistem 29
Gambar 5.5 Implementasi rangkaian sensor 30
Gambar 5.6 Kode program pembacaan sensor untuk mengetahui jarak ideal 30
Gambar 5.7 Kode program pembacaan sensor untuk mendeteksi api 31
Gambar 5.8 Cakupan sensor dan jarak yang ideal dalam mendeteksi api 32
Gambar 5.9 *Pinout mainboard AR Drone 2.0* 33
Gambar 5.10 *Node.JS* pada sistem *quadcopter* 34
Gambar 5.11 *Node* pada sistem *quadcopter* 34
Gambar 6.1 Kode program pembacaan sensor untuk mendeteksi api 36
Gambar 6.2 Tampilan pembacaan sensor pada *Arduino IDE* 37
Gambar 6.3 Tampilan sistem *quadcopter* 38
Gambar 6.4 Kode program untuk mengeksekusi *API* 38

Gambar 6.5 Tampilan hasil dari pembacaan sensor pada sistem *quadcopter*..... 39
Gambar 6.6 Tampilan hasil dari pembacaan sensor pada sistem *quadcopter*..... 41
Gambar 6.7 Skenario 1 pengujian pencarian api dengan metode *Grid*..... 44
Gambar 6.8 Skenario 2 pengujian pencarian api dengan metode *Grid*..... 45
Gambar 6.9 Kode proram pengujian sistem 46
Gambar 6.10 Tampilan *terminal* saat *quadcopter* menemukan api..... 47

BAB 1 PENDAHULUAN

1.1 Latar Belakang

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 saat berada di daerah yang bersuhu dingin. Namun, banyak musibah yang terjadi apabila api tidak dapat dikendalikan. Contohnya yang sering terjadi dalam kehidupan sehari-hari adalah kompor yang meledak dan percikan api pada kabel listrik. 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 *autonomus* 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 penulis akan 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 Nano*. 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* (Barrow, 2014). Metode pencarian api yang digunakan dalam penelitian ini adalah metode *Grid*. Pencarian api dengan metode *grid* pada penelitian ini menggunakan 2 skenario pencarian, skenario pertama api diletakkan pada posisi terjauh dari *quadcopter,* dan skenario kedua api diletakkan pada posisi terdekat dari *quadcopter*.

1.2 Rumusan Masalah

Seperti yang sudah diuraikan diatas, maka perumusan masalah yang akan dibahas pada penelitian ini adalah sebagai berikut:

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

1.3 Tujuan

Tujuan dari penelitian ini adalah :

1. Penggunaan sensor api *UVTRON* untuk mendeteksi api.
2. Melakukan pencarian api secara *autonomous* menggunakan *quadcopter.*
3. Membangun sistem pendeteksi api yang terjangkau dengan menggunakan *Quadcopter AR Drone 2.0.*

1.4 Manfaat

Manfaat dari penelitian ini adalah :

1. Bagi pemadam kebakaran atau dinas kehutanan dapat mendeteksi titik api yang menyala tanpa harus penerjunan manusia ke lokasi kebakaran.
2. Bagi para peneliti, hasil penelitian dapat dikembangkan menjadi sebuah penelitian baru.
3. Mengetahui metode pencarian api yang akurat.

1.5 Batasan Masalah

Adapun batasan masalah agar tidak menyimpang dari perumusan masalah sebagai berikut :

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.

1.6 Sistematika Pembahasan

Sistematika penulisan bertujuan sebagai penjelasan umum dari bagian-bagian bab yang ada dalam penelitian ini agar memudahkan pembaca dalam mengikuti alur pembahasan penelitian. Adapun sistematika penulisan adalah sebagai berikut:

1. BAB I: PENDAHULUAN

Pada BAB I memuat latar belakang permasalahan, rumusan masalah, identifikasi dan pembatasan masalah, tujuan dan manfaat, dan sistematika penulisan.

2. BAB II: DASAR TEORI

Pada BAB II berisi penjelasan dari teori-teori dasar yang menjadi acuan dalam melakukan penerapan penelitian sensor ultrasonik untuk menghindari penghalang pada quadcopter, beberapa penjelasannya dikutip dari berbagai sumber literature seperti jurnal dan paper.

3. BAB III: PEMBAHASAN

Pada BAB III berisi penjelasan mengenai metode penelitian dari penelitian yang akan dilakukan, seperti studi literature, analisis kebutuhan, pengumpulan data, pengolahan data, desain sistem yang akan dibangun, implementasi, pengujian, analisis dan kesimpulan dari penelitian yang akan dilakukan.

4. BAB IV: PEREKAYASA KEBUTUHAN

Pada BAB IV berisi penjelasan mengenai kebutuhan-kebutuhan yang terkait dalam penelitian seperti kebutuhan user dan kebutuhan pengguna, serta kebutuhan fungsional dan non fungsional.

5. BAB V: PERANCANGAN DAN IMPLEMENTASI

Pada BAB V berisi penjelasan perancangan sistem seperti diagram blok dan diagram alur kerja sistem.

6. BAB VI: PENGUJIAN

Pada BAB VI berisi penjelasan mengenai proses pengujian dimulai dari tujuan pengujian, prosedur pengujian, pelaksanaan pengujian, hasil pengujian dan dilakukan analisa hasil pengujian yang sudah dilakukan.

1.7 Jadwal Penelitian

Tabel 1.1 Jadwal Penelitian

No.	Kegiatan	Bulan ke-			
		1	2	3	4
1	Studi Literatur				
2	Perancangan				
3	Implementasi				
4	Pengujian dan Analisa				
5	Penarikan Kesimpulan				

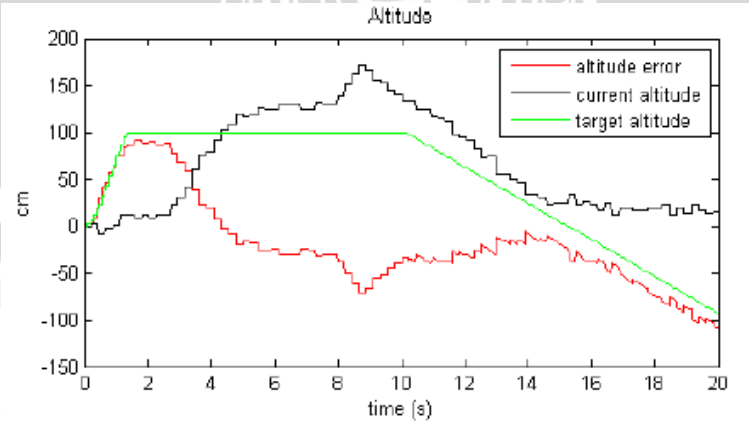
BAB 2 LANDASAN KEPUSTAKAAN

Pada bab ini akan menjelaskan tentang tinjauan pustaka, dasar teori dan algoritma yang akan digunakan dalam penelitian.

2.1 Tinjauan Pustaka

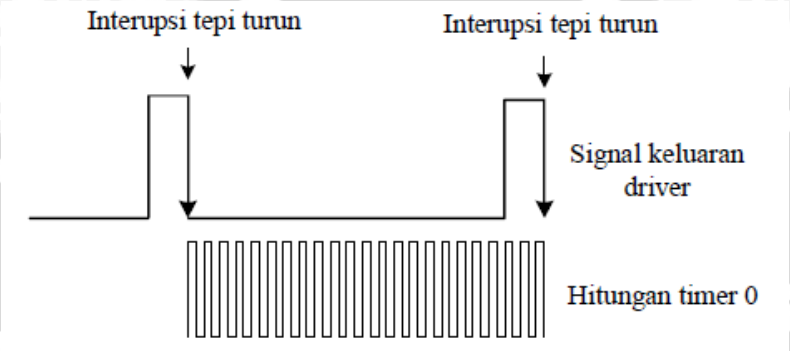
Tinjauan pustaka berisi tentang penelitian yang sudah dilakukan sebelumnya dan berkaitan dengan penelitian yang akan dilakukan. Tujuan dengan adanya tinjauan pustaka yang sudah ada agar dapat dijadikan perbandingan dalam pelaksanaan penelitian.

Pada penelitian yang berjudul *Autonomous Quadcopter for Fire Detection* (Chang. et al, 2015) menerapkan teknik sensor panas dan sensor gas untuk mendeteksi api. *Quadcopter* dirancang untuk mendeteksi api berdasarkan suhu dari suatu benda dan asap berupa CO_2 yang dihasilkan dari pembakaran. Pada saat *"DISARMED"*, *Quadcopter* tidak akan bergerak sampai user mengirimkan perintah untuk *takeoff*. Ketika *Quadcopter* telah mencapai ketinggian yang stabil maka kondisi *Quadcopter* akan berubah menjadi kondisi "*FLY*". Pada kondisi *"FLY"* ini maka *Quadcopter* akan mulai mencari tanda dari api. Jika api atau sumber panas lainnya terdeteksi, kondisi pada *Quadcopter* akan menjadi "*FIRE*" dimana *Quadcopter* akan mengitari dan mencari sumber panas. Pada saat sumber panas telah ditemukan, *Quadcopter* akan memberikan pesan ke komputer dan GPS akan melokasikan titik api tersebut berada dan jika api bergerak, *Quadcopter* akan mengikuti pergerakan tersebut. Pada gambar 2.1 menunjukkan pada sepuluh detik pertama adalah kondisi *"Takeoff"* dan sepuluh detik terakhir kondisi *landing*. Penelitian ini menjadi acuan dalam penentuan ketinggian yang optimal dari *Quadcopter* menuju titik api yang diterapkan dalam *AR Drone 2.0.*



Gambar 2.1 Kondisi ketinggian
Sumber: Chang. et al (2015)

Pada penelitian kedua yang dijadikan tinjauan pustaka berjudul Analisis dan Desain Penginderaan Sensor Api *Hamamatsu UVTRON R2868* (Pracoyo, 2012). Penelitian ini melakukan pengembangan dari sensor *UVTRON* R2868 yang dapat menggambarkan penginderaan dengan membuat *library* untuk membaca data pulsa dari sensor *UVTRON* menjadi data *binary* atau desimal. Dengan konversi data analog ke digital maka akan memberikan informasi sensitivitas jarak pembacaan sensor dan mengindra perubahan jarak api terdekat.



Gambar 2.2 Konversi data pulsa ke desimal
Sumber: Pracoyo (2012)

Setiap tepi turun dari keluaran UVTRON melakukan *STOP timer,* pengambilan *register TCNT* 0 dikali dengan jumlah *overflow* yang terjadi. Hal ini dilakukan secara periodik setiap sensor menangkap api. Isi register TCNT dikali dengan jumlah *overflow* adalah representasi dari hasil konversi intensitas sinar UV. Penelitian ini dijadikan bahan studi pustaka untuk mendapatkan nilai sensitivitas sensor dan resolusi sinar UV untuk melihat pengujian sensor yang dilakukan.

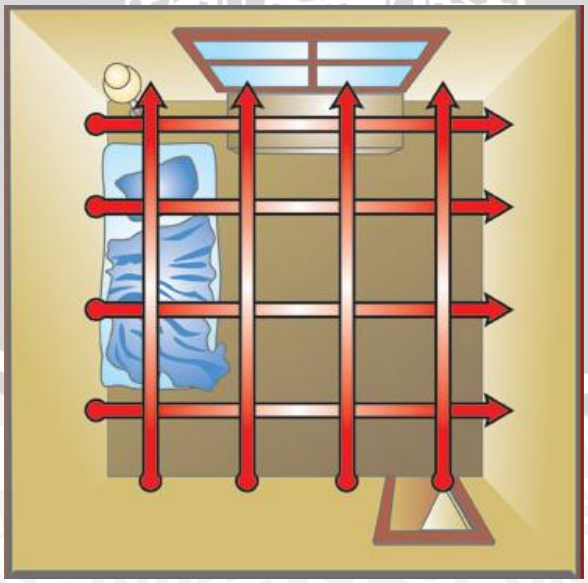
Selain mengkaji dari hasil kedua penelitian yang telah disebutkan, penulis juga akan mengkaji dari *project* Erik Barrow yang berjudul *Autonomous Navigation and Search in an Indoor Envirenment Using an AR Drone* yang dimana penulis akan menerapkan metode pencarian untuk menemukan titik api didalam ruangan *(Indoor)*. Sistem ini diharapkan mampu melakukan pencarian api dengan metode *grid.* Metode pencarian *grid* sendiri akan membutuhkan waktu dalam pencarian, namun pada pencarian ini sangat detail untuk menemukan api dibandingkan metode yang lainnya seperti *spiral, zone* dan *parallel* untuk lebih lengkapnya terdapat pada gambar 2.3.

Algorithm	Advantages	Disadvantages
Spiral	❖ Large area covered ❖ Quick ❖ Covers all axes	❖ Best if robot is centred ❖ May miss sections of room when encountering walls
Grid	❖ Large area	❖ Long time to fly entire grid
Zone	❖ Quick –less area to cover ❖ Cuts down search space ❖ Will deal with expanding rooms well	❖ Less accurate as covering larger quadrants
Parallel	❖ Fast, only has a set number of movement to do	❖ Only on one axis, camera may miss object. ❖ Less advanced version of grid

Gambar 2.3 Perbandingan Metode Pencarian
Sumber: Barrow (2014)

Berdasarkan gambar tersebut didapatkan data bahwa:

1. *Spiral* : Pencarian ini cepat, luas namun robot harus berada ditengah area pencarian dan melewatkan pencarian pada bagian sudut ruangan.
2. *Grid* : Area yang luas namun membutuhkan waktu dalam pencarian dibandingkan pencarian lainnya.
3. *Zone* : Cepat namun area yang ditelusuri sedikit dan menyisakan banyak area yang tidak dijangkau.
4. *Parallel* : Cepat dengan beberapa gerakan yang harus dilakukan. Pencarian satu arah yang memungkinkan beberapa area terlewati.



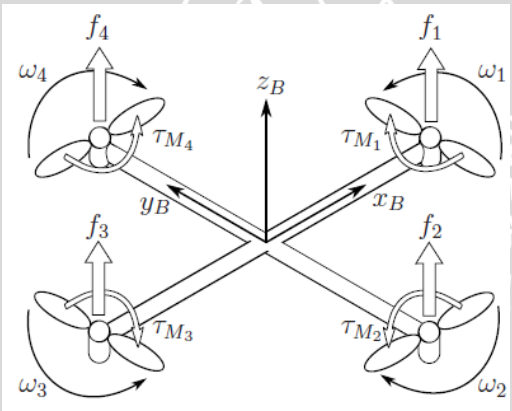
Gambar 2.4 Metode pencarian *Grid*
Sumber: Barrow (2014)

2.2 Dasar Teori

Dasar teori diperlukan untuk membahas teori yang berkaitan dengan penelitian yang dikerjakan.

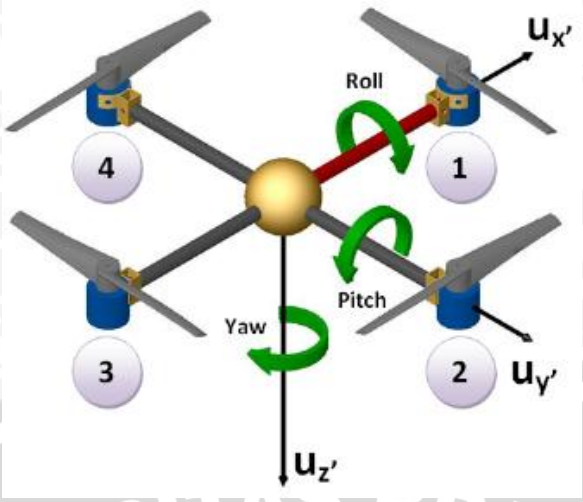
2.2.1 *Unmanned Aerial Vehicle* (UAV) *Quadcopter*

Quadcopter atau yang dikenal juga dengan nama *quadrotor*, adalah sebuah helikopter dengan empat rotor. Rotor-rotor diarahkan menghadap atas dan ditempatkan dengan formasi membentuk persegi dengan jarak yang sama tiap rotornya menuju titik tengah dari *quadcopter* tersebut. *Quadcopter* dikendalikan dengan penyesuaian kecepatan sudut dari tiap rotor yang diputar dengan tenaga motor listrik. *Quadcopter* merupakan desain UAV kecil karena memiliki struktur yang sederhana (Luukkonen, 2011). *Quadcopter* digunakan untuk pengawasan, pencarian dan penyelamatan, inspeksi dan pengaplikasian lainnya. Karena tergolong UAV berukuran kecil maka banyak kalangan menggunakan *Quadcopter* sebagai bahan penelitian.

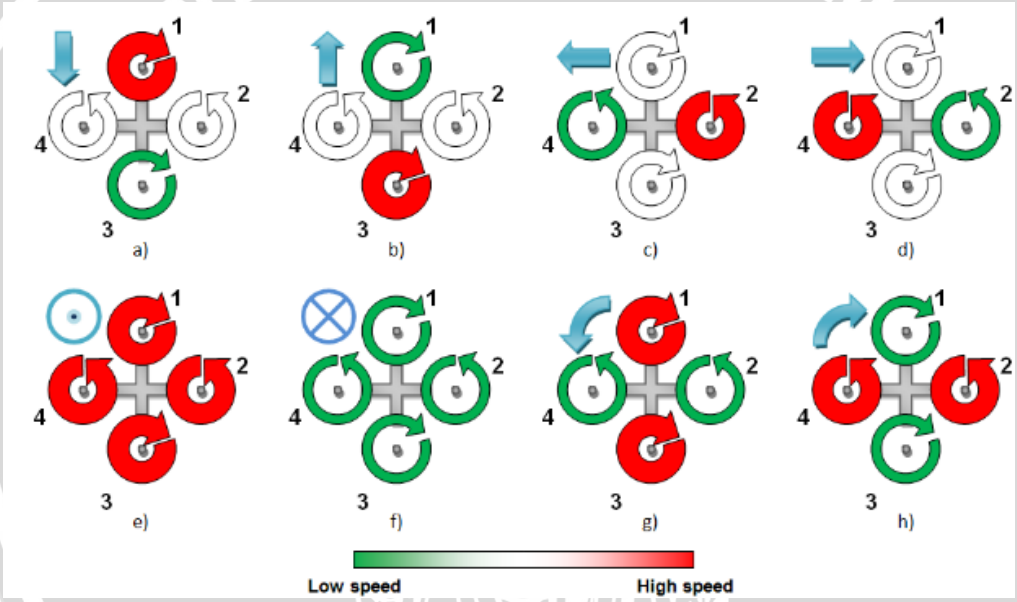


Gambar 2.5 Rangka *Quadcopter*
Sumber: (Luukkonen, 2011)

Tiap rotor pada *quadcopter* memiliki fungsi sebagai pendorong dan rotor pada *quadcopter* ini tidak semuanya berputar searah jarum jam. Rotor pada *quadcopter* dibagi dalam dua pasang, dua pendorong dan dua baling penarik yang bergerak berlawanan arah jarum jam. Dari pembagian ini, torsi yang dihasilkan dapat mengangkat badan *quadcopter.* Untuk orientasi terbang *quadcopter* sendiri terdapat tiga parameter yaitu sudut *yaw, pitch* dan *roll*. Untuk ilustrasi dari orientasi dan parameter *quadcopter* dapat dilihat pada gambar 2.4.



Gambar 2.6 *Yaw, pitch dan roll*
Sumber: (Jorge, 2009)



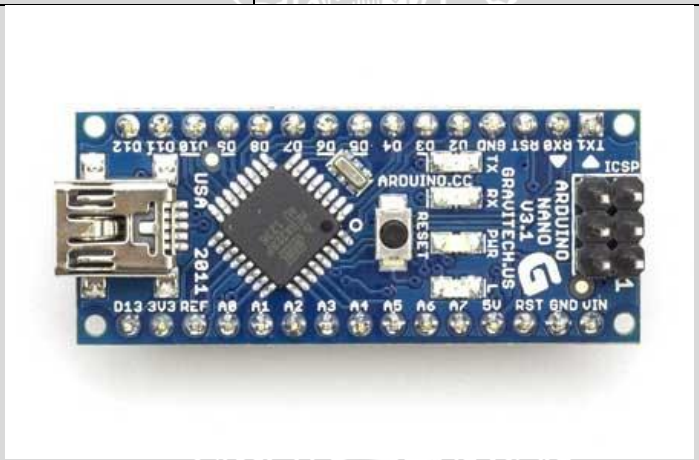
Gambar 2.7 Ilustrasi pergerakan Rotor
Sumber: (Jorge, 2009)

2.2.2 *Arduino Nano*

Pada penelitian ini mikrokontroler yang digunakan adalah *Arduino Nano* (Gambar 2.6). *Arduino Nano* merupakan jenis mikrokontroler *Open Source* yang banyak digunakan dalam penelitian serta pembuatan prototipe produk. *Arduino Nano* mudah digunakan karena tidak butuh menggunakan sambungan *FTDI* dan berukuran kecil sehingga tidak terlalu memakan tempat. Spesifikasi dari *Arduino Nano* terdapat pada tabel 2.1

Tabel 2.1 Tabel spesifikasi *Arduino Nano*

Mikrokontroler	ATmega328
Tegangan pengoperasian	5 V
Tegangan *input* yang disarankan	7-12 V
Batas tegangan *input*	6-20 V
Jumlah pin I/O digital	14 (Memiliki Pin *output* sebanyak 6 buah)
Jumlah pin *input* analog	8
Arus DC tiap pin I/O	40 mA
Arus DC untuk pin 3.3V	50 mA
Memori Flash	32 KB (ATmega328), 2 KB *bootloader*
SRAM	2 KB
EEPROM	1 KB
Clock Speed	16 MHz



Gambar 2.8 *Arduino Nano*
Sumber: Arduino.cc (2016)

2.2.3 *Parrot AR Drone 2.0*

Parrot AR Drone adalah *quadcopter* yang digunakan dalam penelitian ini. *Parrot AR Drone* sendiri menggunakan *mainboard* berjenis *AR Drone 2.0* versi 2. Spesifikasi dari *mainboard* ini menggunakan prosessor *Parrot* P6 (32bits ARM9core, dan berjalan pada frekuensi 468 MHz). *AR Drone 2.0* ini dilengkapi dengan modul *Wi-Fi* dan dua modul kamera yang berada dibagian depan dan bagian bawah. Untuk sistem operasi yang digunakan oleh *mainboard AR Drone 2.0* adalah *Linux* berbasis *real-time operating system* dimana semua proses diolah dan dijalankan pada prosessor P6. Spesifikasi kamera depan yang terdapat pada *board* ini adalah memiliki resolusi sebesar 640 x 480 dengan jumlah *frame rate* sebesar 15 *frame per second.* Kamera pada bagian bagah berfungsi untuk

mengukur kecepatan dari *quadcopter.* Spesifikasi kamera bagian bawah menggunakan lensa kamera diagonal sebesar 64 derajat dengan *frame rate* sebesar 60 *frame per second.* Terdapat pula *port USB* yang dapat digunakan untuk melakukan *flashing* sistem operasi atau perangkat tambahan seperti *GPS* maupun *board* yang menggunakan *USB port.*

Quadcopter ini menggunakan protokol komunikasi UDP agar dapat dikendalikan dengan perintah *Telnet*. Selain menggunakan *Telnet, quadcopter* ini juga dapat dikendalikan menggunakan aplikasi pada perangkat *smartphone* dengan mengunduh aplikasi "*AR Flight Control"* pada perangkat *smartphone*.

Tabel 2.2 Tabel spesifikasi *mainboard AR Drone 2.0*

Prosesor	1 GHz 32-bit *ARM Cortex* dengan 800 MHz *video DSP TMS320DMC64x*
Sistem Operasi	*Linux 2.6.32*
RAM	1 GB DDR2 RAM berkecepatan 200 MHz
USB	USB 2.0 *High Speed*
Konektivitas	*Wi-Fi b, g, n*
Gyroscope	3 *axis gyroscope* 2000° / *seconds precision*
Accelerometer	3 *axis accelerometer +-50mg precision*
Magnetometer	3 *axis magnetometer 6° precision*
Pressure Sensor	*Pressure sensor +/-10 Pa precision*
Ultrasound Sensor	*Ultrasound sensors* untuk perhitungan darat
Camera	60 fps vertikal QVGA kamera untuk pengukuran kecepatan darat

Tabel 2.3 Tabel spesifikasi motor *AR Drone 2.0*

Tipe Motor	4 *brushless inrunner motors.* 14.5 W 28.500 RMP
Bearing	*Micro ball bearing*
Propeller	*Low noise Nylatron gears for 1/8.75 propeller reductor*
Material	*Tempered steel propeller shaft*
CPU	8 *MIPS AVR CPU* per motor *controller*

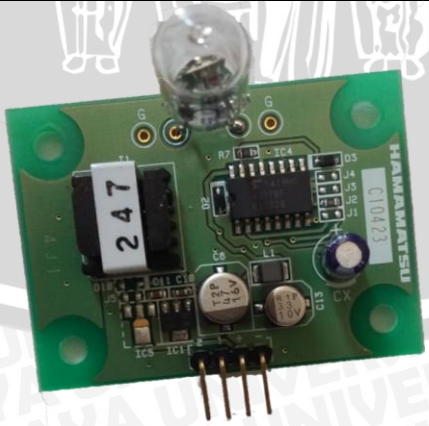
Gambar 2.9 *Parrot AR Drone 2.0*
Sumber: Ardrone2.parrot.com (2016)

2.2.4 Sensor Api UVTRON R9454

Sensor yang digunakan pada penelitian ini adalah sensor Api UVTRON R9454. Sensor api dapat digunakan untuk mendeteksi sinar UV yang ditimbulkan oleh api. Sensor ini merupakan sensor api yang sensitif terhadap api sampai jarak 5 meter. Fungsi dari sensor ini adalah untuk membaca sinar gelombang UV dari api yang menyala dengan panjang gelombang 185 nm sampai dengan 260 nm. Sensor ini dipilih karena jangkauan yang luas dibandingkan sensor TPA81 yang hanya sampai 2 meter.

Tabel 2.4 Tabel spesifikasi sensor *UVTRON R9454*

Spectral Response	185 sampai 260 nm
Tegangan *Input*	12 - 24 V
Arus DC	4 mA
Suhu Operasional	-10 sampai +50 $^{\circ}C$



Gambar 2.10 Sensor api UVTRON R9454
Sumber: Dokumentasi Penulis

1. Pin 1 bertanda “2”. Pin ini adalah *Open collector output* berimpedansi rendah dengan kondisi *Active Low.*
2. Pin 2 bertanda “1”. Pin ini adalah *Open collector output* berimpedansi tinggi dengan kondisi *Active High*.
3. Pin 3 bertanda “+”. Pin ini adalah Catu daya DC 12v.
4. Pin 4 bertanda “-“. Pin ini adalah *Ground*.

Sensor ini membutuhkan tegangan catu sebesar 12V-24V. *Output* yang dihasilkan oleh modul *UVTron* ini berupa pulsa persegi yang periodanya tergantung dari intensitas cahaya ultraviolet yang ditangkapnya. Perioda maksimumnya mencapai 10 ms .Sinyal *output* modul UVTron tersedia dalam dua bentuk, yaitu *output* berimpedansi tinggi dengan kondisi *Active High*, dan *output* berimpedansi rendah dengan kondisi *Active Low*.

2.2.5 *Application Programming Interface (API)*

API adalah sekumpulan perintah, fungsi, serta protokol yang dapat digunakan untuk mempermudah dalam membangun perangkat lunak untuk berinteraksi dengan sistem operasi tertentu dengan menggunakan fungsi standar yang ada. Pada penelitian ini *API* yang digunakan sebagai berikut.

2.2.5.1 Node.JS

Node.JS adalah *runtime JavaScript* yang dibangun pada *Chrome V8 JavaScript Engine. Node.JS* dibuat dengan tujuan agar *JavaScript* dapat berjalan diberbagai platform tidak hanya pada *browser* saja. *Package* pada *Node.JS* bersifat *open source.*

2.2.5.2 Node AR Drone

API pada *Node AR Drone* menggunakan bahasa *Java*. *API* ini digunakan untuk membantu *user* dalam pengembangan pada *mainboard AR Drone 2.0.AR Drone 2.0 API* digunakan untuk menghubungkan program yang telah ditulis *user* dengan *mainboard AR Drone 2.0* seperti perintah *take-off, landing* atau mendapatkan data sensor yang terdapat pada *quadcopter*.

2.2.5.3 Node Serialport

Node Serialport adalah perantara antara hardware sensor yaitu *Arduino* dan *mainboard AR Drone 2.0* sebagai pembaca *JavaScript* agar dapat saling berkomunikasi melalui komunikasi serial (thinkingonthinking, 2016).

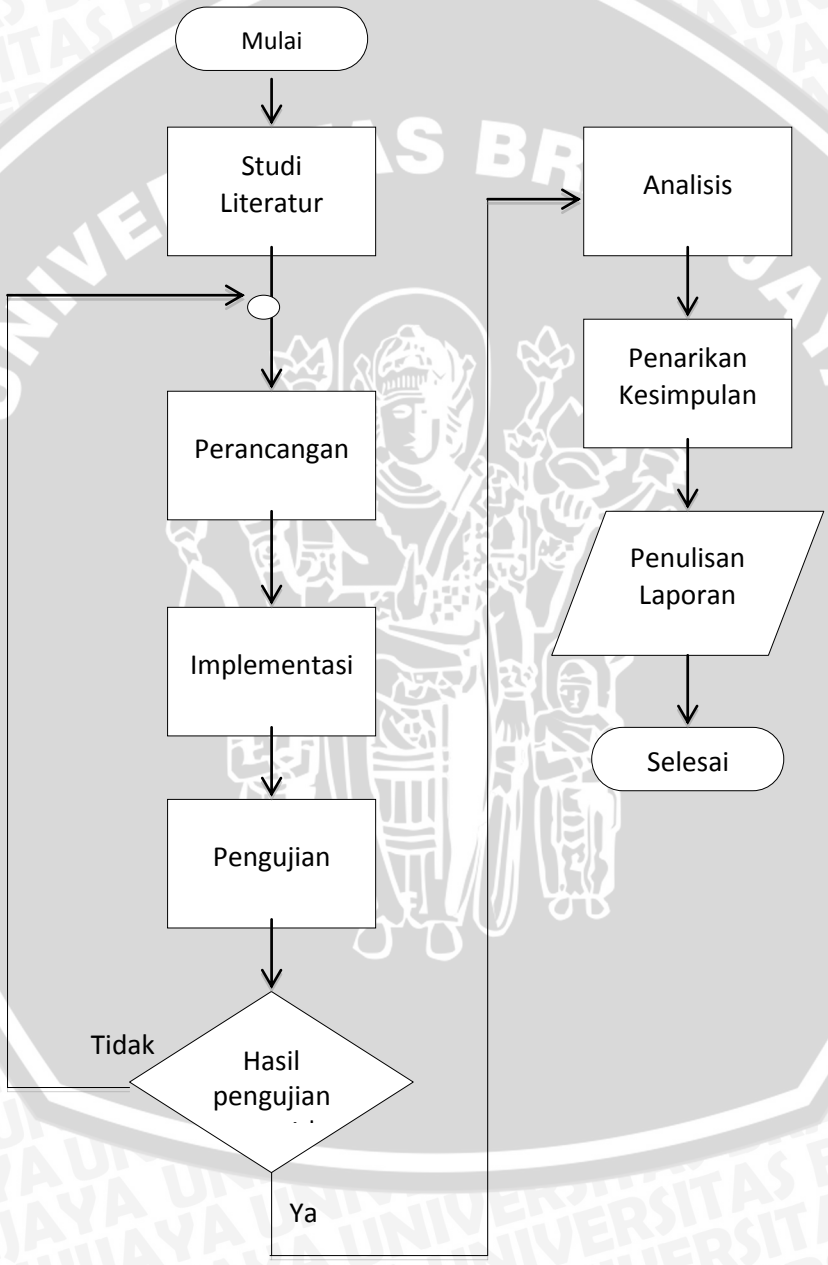
2.2.5.4 Node Package Manager (NPM)

Node Package Manager mempermudah para pengembang *JavaScript* dalam membangun suatu sistem dengan berbagi *code* yang dapat digunakan untuk memecahkan masalah.

BAB 3 METODOLOGI PENELITIAN

3.1 Metode Penelitian

Pada bab ini akan menjelaskan tentang metodologi penelitian yang akan digunakan dalam melakukan penelitian mapun dalam penulisan skripsi. Adapun gambaran umum tahapan-tahapan metodologi penelitian dapat dilihat dari diagram alir sebagai berikut:



Gambar 3.1 Diagram alir penelitian

3.1.1 Studi Literatur

Pada bagian ini dibahas mengenai dasar-teori yang mendukung penelitian perancangan *quadcopter* untuk mendeteksi api dengan metode *grid*. Adapun beberapa dasar-teori yang dijadikan bahan studi dalam penelitian yaitu:

a. *Unmanned Aerial Vehicle (UAV)*

Studi literatur yang terkait dengan penjelasan dasar mengenai *UAV* berjenis *quadcopter* atau *quadrotor*. Mengetahui cara kerja *quadcopter,* pengendalian *quadcopter*, dan kegunaan *quadcopter.*

b. Mikrokontroler

Penjelasan dasar terkait dengan mikrokontroler yang digunakan dalam penelitian ini yaitu *Arduino Nano* dan *mainboard AR Drone 2.0* berdasarkan literatur yang ada.

c. Sensor Api *UVTRON*

Kajian dari studi literatur yang didapat dari jurnal penelitian yang terkait tentang pengembangan dari sensor api *UVTRON* yang diterapkan pada *quadcopter* untuk mendeteksi api sebagai referensi dalam penelitian ini.

d. *Node.JS Node AR Drone*

Referensi tentang Node.JS *node AR Drone 2* agar dapat dimanfaatkan dalam penelitian ini. Referensi didapatkan dari dokumentasi *online* dan petunjuk pengembangan dalam menggunakan *Node.JS node AR Drone.*

e. *Node Serialport*

Referensi terkait dengan *node serialport* didapatkan dari dokumentasi *online* dalam menggunakan *node serialport* agar *mainboard Ar Drone* dapat berkomunikasi dengan board lainnya secara serial.

f. *AR Drone 2.0 API*

API sangat penting pada penelitian ini, dengan perintah, fungsi serta protokol pada *API* yang terkait dengan *quadcopter* untuk mempermudah dalam pengerjaan penelitian.

3.1.2 Rekayasa Persyaratan

Analisa kebutuhan diperlukan untuk menganalisa beberapa kebutuhan yang dibutuhkan sistem pada penelitian yang dilakukan agar sesuai dengan yang dikehendaki. Ada dua jenis analisa kebutuhan yang digunakan dalam penelitian, yaitu analisa kebutuhan pengguna dan analisa kebutuhan sistem.

3.1.2.1 Kebutuhan Pengguna

Berikut merupakan kebutuhan pengguna pada penelitian yang dilakukan, yaitu:

1. Sistem pada *quadcopter* dapat menerima perintah dari komputer *user* sehingga dapat terbang secara *autonomous* serta dapat menerima dan memahami perintah berdasarkan hasil pembacaan sensor *UVTRON* sehingga *quadcopter* dapat mendeteksi api.

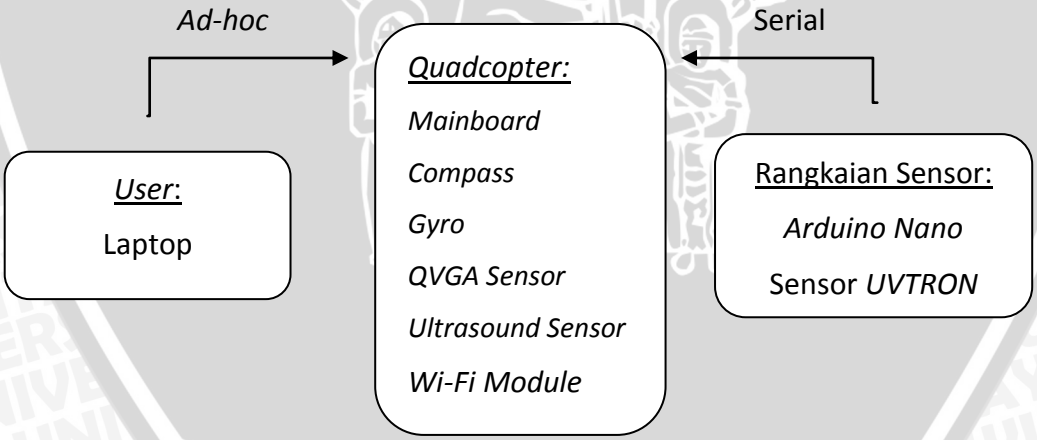
3.1.2.2 Kebutuhan Sistem

Terdapat beberapa kebutuhan sistem yang diperlukan pada perancangan sistem dalam penelitian, yaitu:

1. Sistem *quadcopter* terhubung ke komputer *user* secara *ad-hoc.*
2. Komputer *user* dapat masuk ke dalam sistem *quadcopter* dan mengirimkan perintah melalui *telnet.*
3. Serial port pada *mainboard* mampu mendapatkan dan membaca data sensor dengan menggunakan *API node serialport.*
4. Data hasil pembacaan sensor dapat dikirimkan melalui komunikasi serial antara *Arduino* dan *mainboard quadcopter.*
5. Sistem *quadcopter* mampu mengubah data sensor yang didapatkan menjadi *command* menggunakan *Node.JS* sehingga *quadcopter* dapat mendeteksi adanya api.

3.1.2.3 Kebutuhan *Hardware*

Berikut adalah kebutuhan *hardware* yang digunakan pada perancangan sistem dalam penelitian, yaitu pada gambar 3.2.



Gambar 3.2 Kebutuhan *Hardware*

Berikut penjelasan lebih rinci terkait kebutuhan *hardware* dari sistem yaitu:

1. *User*

   a. Pada bagian ini Laptop *user* digunakan untuk menuliskan dan mengirimkan perintah melalui koneksi *Ad-hoc* agar *quadcopter* dapat terbang secara *autonomous.*

2. *Quadcopter*

   a. *Mainboard AR Drone 2.0* berfungsi sebagai pusat kendali komponen yang ada pada *quadcopter*. *Mainboard* ini memiliki sistem operasi *linux BusyBox v.1.14.0* yang dapat diakses secara *telnet* melalui komputer *user.*

   b. *Compass* merupakan navigasi pada sistem *quadcopter* untuk menentukan orientasi *quadcopter.*

   c. *Gyro* adalah penghitung nilai *roll, pitch* dan *yaw.* Fungsinya pada *quadcopter* sebagai penyeimbang kestabilan terbang.

   d. *QVGA* Sensor pada *quadcopter* berfungsi untuk mengukur kecepatan dipermukaan tanah.

   e. *Ultrasound Sensor* berfungsi mengukur ketinggian dari permukaan disaat terbang.

   f. *Wi-Fi Module* berfungsi sebagai media komunikasi antara *quadcopter* dengan komputer.

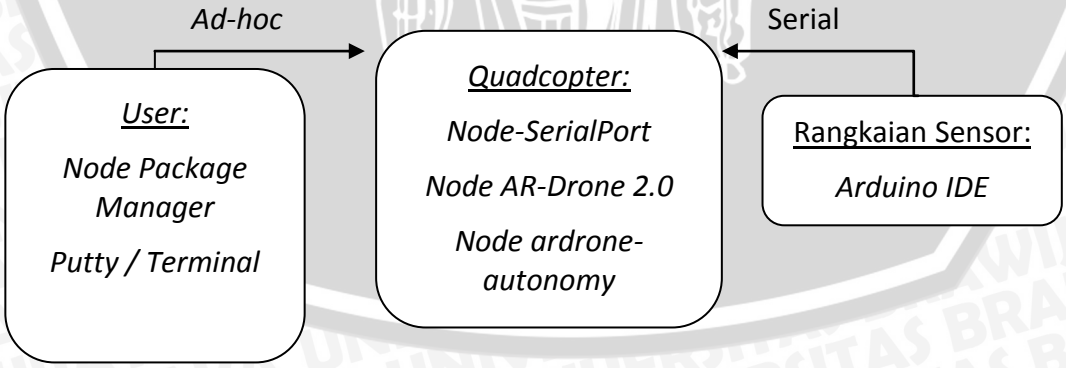
3. Rangkaian Sensor

   a. *Arduino Nano* pada rangkaian ini digunakan sebagai penghubung sensor dan *mainboard* secara serial.

   b. Sensor *UVTRON* merupakan komponen utama pada sistem ini untuk mendeteksi api. Penempatan sensor vertikal menghadap bagian bawah berfungsi untuk mendeteksi api di daratan.

3.1.2.4 Kebutuhan *Software*

Pada bagian ini akan menjelaskan kebutuhan *software* yang diperlukan dalam melakukan penelitian, yaitu pada gambar 3.3.



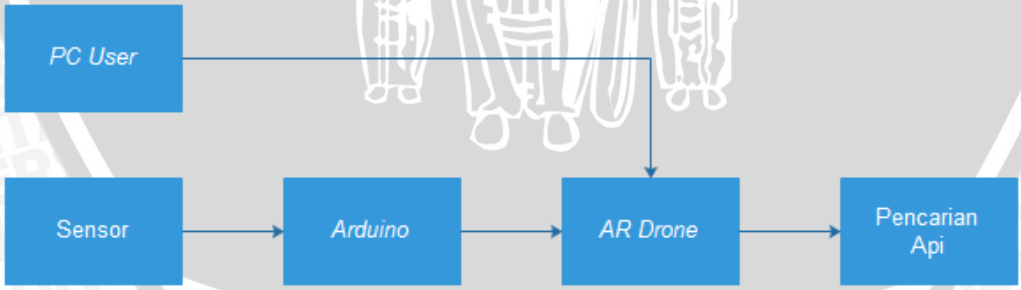
Gambar 3.3 Kebutuhan *Software*

Berikut penjelasan lebih rinci terkait kebutuhan *software* dari sistem yaitu:

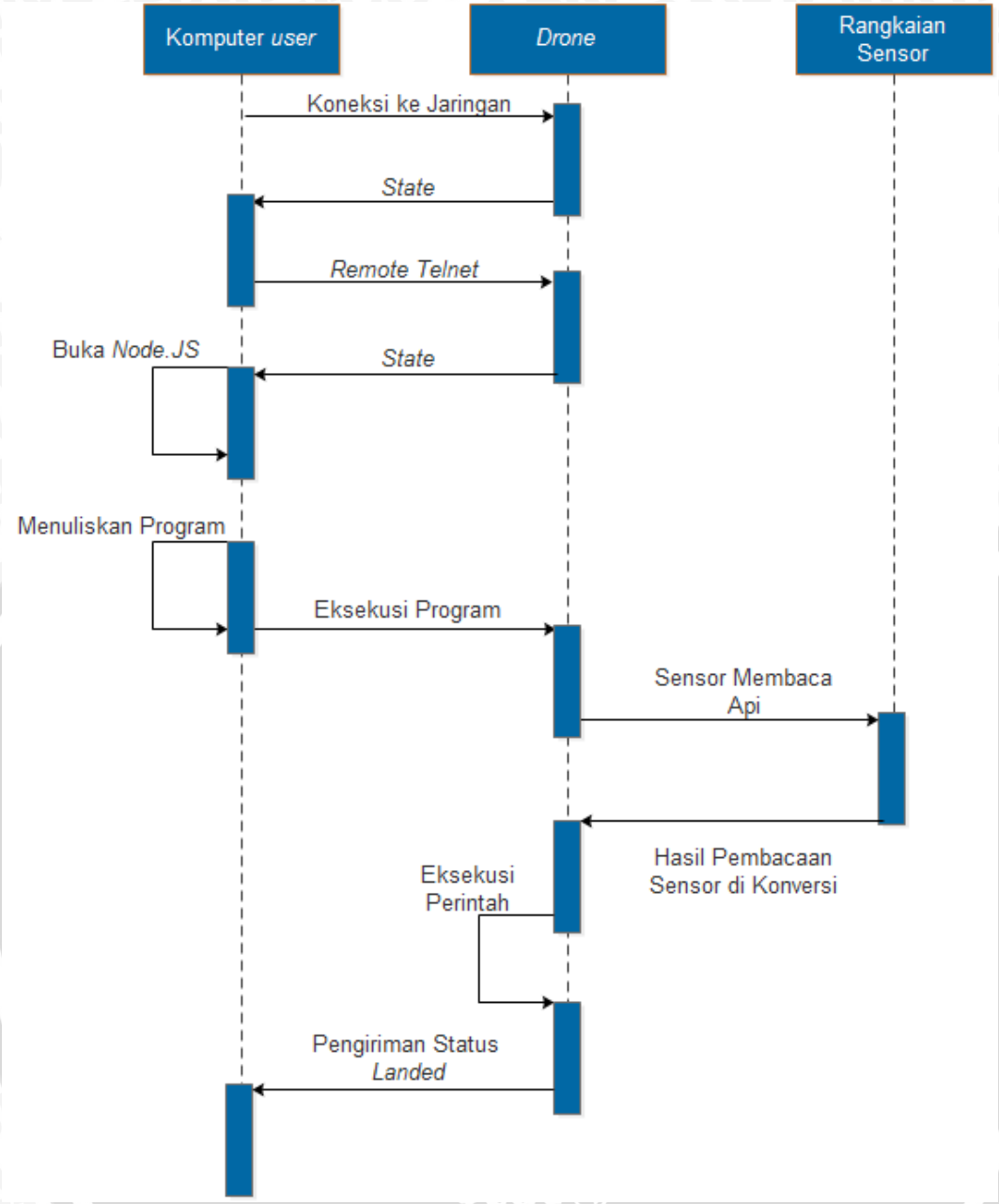
1. *User*
   a. *Node Package Manager* mempermudah para pengembang *JavaScript* dalam membangun suatu sistem dengan berbagi *code* yang dapat digunakan untuk memecahkan masalah.
   b. *Putty / Terminal* digunakan untuk melakukan remote *telnet* dari komputer *user* ke sistem pada *quadcopter*.
2. *Quadcopter*
   a. *Node-SerialPort* berperan sebagai API yang berfungsi menjembatani *hardware* sensor yaitu *Arduino* dan *mainboard Ar Drone* secara komunikasi serial.
   b. *Node AR-Drone* digunakan untuk mengubah data sensor yang didapat menjadi *command* sehingga *quadcopter* dapat mendeteksi adanya api.
   c. *Node ardrone-autonomy* digunakan untuk memberikan perintah terbang sehingga *quadcopter* dapat terbang sesuai dengan kehendak *user.*
3. Rangkaian Sensor
   a. *Arduino IDE* berfungsi sebagai uploader kode algoritma yang telah dibuat sebelumnya.

3.1.3 Perancangan Sistem

Perancangan sistem merupakan tahapan untuk membangun sistem dari penelitian setelah melakukan tahapan studi literatur, analisa dan tahapan perekayasaan sistem agar penelitian dapat dilakukan secara sistematis dan terstruktur. Diagram blok dari perancangan sistem penelitian dapat dilihat pada gambar 3.4 sebagai berikut:



Gambar 3.4 Diagram blok sistem



Gambar 3.5 *Sequence Diagram* Sistem

Berdasarkan *sequence diagram sistem* pada gambar 3.5, alur kerja sistem yang sudah dibuat untuk penelitian ini dapat dijelaskan sebagai berikut:

1. *PC User* dikoneksikan ke jaringan *AR Drone 2.0.*
2. *PC User* melakukan *remote* dengan *telnet* ke *quadcopter* dengan menggunakan alamat 192.168.1.1 port 23.
3. Ketika *user* sudah berhasil masuk kedalam sistem *quadcopter,* maka dapat mengeksekusi *Node.JS* untuk menuliskan kode algoritma melalui *telnet* pada *quadcopter* agar *quadcopter* dapat beroperasi secara

autonomous serta dapat membaca dan mengubah data sensor yang dikirimkan melalui serial.

4. Sensor yang terhubung ke *Arduino* mengirimkan data pembacaan ke *mainboard quadcopter* secara serial.
5. Data sensor yang diterima menggunakan bahasa *javascript* dan akan dieksekusi oleh *quadcopter* sesuai perintah yang dituliskan.

3.1.4 Implementasi Sistem

Pada tahap implementasi sistem ini diawali dengan perancangan rangkaian sensor api seperti letak posisi dari komponen-komponen yang digunakan agar didapatkan rangkaian sensor yang memiliki ukuran dan berat yang efektif untuk dipasangkan pada *quadcopter*. Setelah itu dilakukan penulisan algoritma deteksi api agar rangkaian sensor bekerja sesuai dengan yang diinginkan. Pada tahapan selanjutnya dilakukan pengimplementasian *API* di sistem *quadcopter* yang dapat membantu *user* untuk menuliskan algoritma secara langsung pada sistem *quadcopter,* membaca data sensor secara serial kemudian mengubah data hasil pembacaan sensor menjadi perintah dan mengeksekusi perintah tersebut.

3.1.5 Pengujian dan Analisis

Pada tahap pengujian dan analisis sistem dilakukan sesuai dengan parameter perancangan sistem. Berikut merupakan skenario dalam melakukan pengujian sistem, yaitu:

1. Pengujian deteksi api menggunakan rangkaian sensor.
2. Pengujian penerimaan dan pembacaan data pada sistem *quadcopter* menggunakan *API.*
3. Pengujian fungsi *API Node AR Drone 2.0* agar dapat beroperasi secara *autonomous.*
4. Pengujian waktu respon pergerakan *quadcopter* terhadap titik api.

3.1.6 Penarikan Kesimpulan

Penarikan kesimpulan merupakan tahap yang dilakukan setelah melakukan seluruh pengujian sistem yang telah dirancang. Dari kesimpulan yang sudah didapatkan, kesimpulan dan saran dapat dimanfaatkan untuk acuan dalam pengembangan sistem yang lebih lanjut sehingga penelitian ini tidak berhenti ketika penelitian ini berakhir.

BAB 4 REKAYASA PERSYARATAN

4.1 Pendahuluan

4.1.1 Tujuan

Pada bab ini akan menjelaskan penelitian tentang pengembangan *Unmanned Aerial Vehicle (UAV)* untuk mendeteksi api dengan lebih rinci berdasarkan format *Software Requirement Spesification (SRS)* IEEEE 830. Adapun penjelasan yang lebih rinci terhadap penelitian ini adalah tujuan dan fitur dari sistem yang dikembangkan, cara kerja sistem, batasan tentang bagaimana sistem ketika dijalankan, kebutuhan atar muka sistem serta penjelasan terkait kebutuhan fungsional dan non fungsional. Bab ini dibuat sebagai dokumentasi untuk para pengembang sistem maupun *stakeholder* dan untuk selanjutnya diusulkan sebagai laporan skripsi.

4.1.2 Ruang Lingkup

Tujuan dari sistem yang dikembangkan ini adalah membuat sebuah sistem pada *quadcopter* agar dapat mendeteksi api secara *autonomous.* Sistem yang dikembangkan yaitu rangkaian pendeteksi api dimana terdapat sensor api *UVTRON* dan *Arduino Nano* yang akan menghasilkan data pembacaan dan seterusnya akan dikirimkan secara serial ke *mainboard quadcopter.* Selanjutnya pada sistem *quadcopter* dimasukkan *API* yang mendukung penelitian seperti *Node AR Drone 2.0, Node.JS* dan *Node Serialport* yang masing-masing memiliki peran agar *quadcopter* dapat membaca data yang dikirimkan oleh sensor secara serial, yang kemudian diubah menjadi perintah agar *quadcopter* dapat mendeteksi api. Pengembangan sistem ini dapat dilakukan lebih lanjut bagi para penguji, pengembangan serta *user* agar didapatkan sistem *quadcopter* yang dapat dimanfaatkan untuk membantu kegiatan manusia sehari-hari serta menghasilkan sebuah riset berskala panjang karena saat ini sudah sangat banyak yang melakukan penelitian dibidang *quadcopter* dan sangat banyak yang diteliti.

4.1.3 Istilah

Tabel 4.1 Tabel Istilah

Istilah	Definisi
UAV Quadcopter	Helikopter dengan empat buah rotor tanpa awak
API	*Applications Programming Interface* yaitu kumpulan dari fungsi-fungsi membantu pengembang membangun sebuah aplikasi.
Komunikasi Serial	Komunikasi yang pengiriman datanya per-bit secara simultan.

4.1.4 Sistematika

Agar penelitian dapat dilihat dan dipahami dengan mudah, maka dibuat sistematika yang menjadi kerangka dan pedoman dalam pembuatan SRS. Dalam pembuatan dokumen ini dibagi menjadi tiga bagian, yaitu:

1. Pendahuluan

   Pada bagian pendahuluan ini terdapat tujuan penelitian, ruang lingkup, istilah, referensi dan sistematika penulisan.

2. Deksripsi Umum

   Dibagian deskripsi umum ini tersusun dari perspektif sistem yang dikembangkan, fungsi dari sistem yang dikembangkan, karakteristik pengguna dan lingkungan operasi dari sistem.

3. Spesifikasi Kebutuhan

   Pada spesifikasi kebutuhan ini menjelaskan lebih detail terhadap kebutuhan dari sistem yang dikembangkan seperti kebutuhan antarmuka sistem, kebutuhan fungsional dan kebutuhan non fungsional sistem.

4.2 Deskripsi Umum

4.2.1 Perspektif Sistem

Sistem deteksi api pada *quadcopter* ini dapat berjalan dan beroperasi ketika komputer *user* dapat terkoneksi secara *ad-hoc* dengan *quadcopter* dan dapat masuk kedalam sistem operasi *mainboard* menggukana *telnet* untuk mengakses *Node.JS*. Fungsi mengakses *Node.JS* adalah untuk memasukkan algoritma yang selanjutnya pada sensor data dikirim secara serial ke *mainboard quadcopter*. Pada *mainboard* sendiri harus dapat membaca dan mengubah data sensor menjadi sebuah perintah sesuai dengan algoritma yang sudah dituliskan sebelumnya. Sebelum melakukan langkah-langkah tersebut diharuskan untuk melakukan kalibrasi agar *quadcopter* dapat terbang stabil. Selain itu catu daya harus terisi penuh untuk menghindari kondisi abnormal.

4.2.2 Kegunaan

Dengan adanya sistem deteksi api pada *quadcopter* ini diharapkan menjadi tahap awal dalam riset jangka panjang pengembangan *quadcopter* bagi Fakultas Ilmu Komputer Universitas Brawijaya sehingga didapatkan *quadcopter* yang dapat membantu dalam kehidupan harian manusia dan juga publikasi ilmiah baik nasional maupun internasional. Secara umum didapatkan *quadcopter* dengan harga yang terjangkau dan dapat mendeteksi adanya api.

4.2.3 Karakteristik Pengguna

Pada penelitian ini *user* berperan dalam tahapan proses seperti kalibrasi *quadcopter* untuk menjaga ke stablian dalam pengoperasian dengan menggunakan aplikasi pada smartphone keluaran *Parrot* bernama *AR.FreeFlight*. Kemudian *user* berperan dalam menginputkan algoritma melalui *Node.JS* agar *quadcopter* dapat beroperasi secara *autonomous.*

4.2.4 Lingkungan Operasi

Ada beberapa kondisi lingkungan yang mempengaruhi sistem saat beroperasi, yaitu:

1. Untuk pengoperasian *indoor* kondisi ruangan harus luas dan tidak terlalu banyak barang. Karena kemungkinan *quadcopter* akan menyesuaikan kondisi area untuk *take-off* dan *landing.*
2. Kondisi ruangan harus cukup cahaya.

4.2.5 Batasan Perancangan dan Implementasi

Adapun batasan dalam perancangan dan implementasi sistem ini adalah sebagai berikut:

1. *Quadcopter* hanya dapat mendeteksi api pada posisi bawah karena penempatan sensor menghadap arah bawah.
2. Sistem diuji menggunakan skenario membentuk *grid* dengan penempatan lilin pada salah satu sudut area pengujian.

4.2.6 Asumsi dan Ketergantungan

Sistem yang dikembangkan ini memiliki beberapa asumsi dan ketergantungan sebagai persyaratan sistem yaitu sebagai berikut:

1. Pin sensor harus dipasang sesuai skematik pada *datasheet* agar berfungsi.
2. Pada sistem operasi *quadcopter* harus ditambahkan *API Node.JS, Node Serialport* dan *Node AR Drone 2.0.*
3. Untuk masuk ke sistem operasi *quadcopter* harus dilakukan menggunakan *telnet* dengan alamat 192.168.1.1 port 23.

4.3 Kebutuhan Antar muka Eksternal

4.3.1 Antarmuka Pengguna

Pengguna dapat melakukan *remote* secara telnet kedalam sistem operasi *quadcopter* menggunakan *putty* / terminal dengan antarmuka berupa *Command Line Interface (CLI).*

4.3.2 Antarmuka Perangkat Keras

1. *User*

a. Laptop atau Desktop PC

2. *Quadcopter*

   a. *Parrot Ar Drone 2.0 Elite Edition*

3. Rangkaian Sensor

   a. *Arduino Nano*

   b. Sensor *UVTRON R9454*

4.3.3 Antarmuka Perangkat Lunak

1. *User*

   a. *Node Package Manager*

   b. *Putty / Terminal*

2. *Quadcopter*

   a. *Node.JS AR Drone 2.0*

   b. *Node Serialport*

3. Rangkaian Sensor

   a. *Arduino IDE*

4.3.4 Kebutuhan Antarmuka Komunikasi

1. Komunikasi Antara *User* dan *Quadcopter*

   Untuk mengkoneksikan *PC User* dengan *Quadcopter* dibutuhkan komunikasi secara *Wireless Ad-hoc*. Setelah *User* terhubung dengan *Quadcopter* maka dilakukan *remote* kedalam sistem operasi *mainboard Quadcopter* dengan menggunakan fitur *Telnet*.

2. Komunikasi antara *Quadcopter* dan Rangkaian Sensor

   Agar rangkaian sensor dapat mengirimkan data dan terhubung ke *mainboard Quadcopter* maka digunakan komunikasi secara serial dengan cara menghubungkan pin *TX (Transceiver)* pada *Arduino* ke pin *RX (Receiver)* pada *mainboard Quadcopter.*

4.4 Kebutuhan Fungsional

Kebutuhan fungsional merupakan pernyataan yang menjelaskan bagaimana sistem dapat memberikan layanan dengan seharusnya sehingga kebutuhan fungsional harus dapat dibenuhi. Adapun kebutuhan fungsional dari sistem yang harus dipenuhi ini adalah sebagai berikut:

1. Fungsi mengakses sistem operasi *Quadcopter* melalui *Telnet*

Sistem operasi *Quadcopter* dapat diakses oleh *user* melalui *Telnet* ke alamat 192.168.1.1, jika berhasil maka *user* akan masuk ke sistem pada *Quadcopter* berupa CLI *(Command Line Interface) BusyBox* versi 1.14.0.

2. Sistem mendeteksi api dan mengirimkan data pembacaan sensor

   Sistem memiliki sensor yang dapat mendeteksi api dan mengirimkan hasil pengolahan data tersebut ke sistem *Quadcopter* menggunakan komunikasi serial.

3. Sistem menerima data sensor dan mengubah data sensor menjadi *command*

   Fungsi ini mengharuskan sistem mengolah data dari sensor menjadi *command* agar *quadcopter* bergerak sesuai dengan perintah yang diprogramkan.

4. Sistem dapat mengeksekusi kode program

   Fungsi ini mengharuskan sistem untuk dapat mengeksekusi kode program yang dituliskan pada *Node.JS*. Fungsi ini sangat penting karena kerja setiap fungsi saling berkesinambungan.

5. Sistem dapat terbang mengikuti perintah terbang

   Fungsi ini mengharuskan sistem untuk dapat terbang secara *autonomous* membentuk *pattern* yang sudah diperintahkan. Pada penelitian ini *pattern* yang digunakan *grid.*

4.5 Kebutuhan Non Fungsional

Kebutuhan Non Fungsional pada sistem ini adalah sebagai berikut:

1. Kebutuhan Performa

   Sistem akan bekerja secara optimal jika daya pada baterai diatas 30%, hal ini dikarenakan jika daya dibawah 30% maka performa terbang *quadcopter* dan komunikasi serial akan menurun dan terganggu.

2. Kebutuhan Keselamatan

   Pada saat menerbangkan *quadcopter* diharuskan untuk menggunakan *indoor hull*, ketika *quadcopter* terbang secara tidak stabil atau ada gangguan badan *quadcopter* tidak akan langsung menyentuh benda keras terlebih dahulu.

BAB 5 PERANCANGAN DAN IMPLEMENTASI

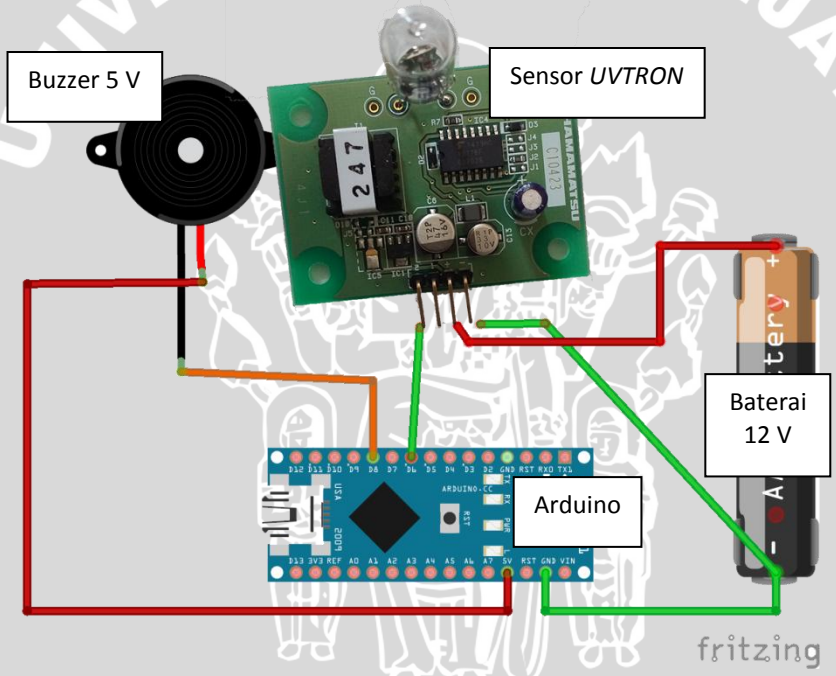
Pada bab ini akan menjelaskan tentang perancangan, cara kerja sistem dan implementasi dari penelitian pengembangan *quadcopter* agar dapat mencari dan mendeteksi api secara detail.

5.1 Perancangan Sistem

Pada bagian perancangan sistem ini terdapat penjelasan mengenai perancangan perangkat keras dan perancangan perangkat lunak.

5.1.1 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 5.1 Skematik rancangan rangkaian sensor

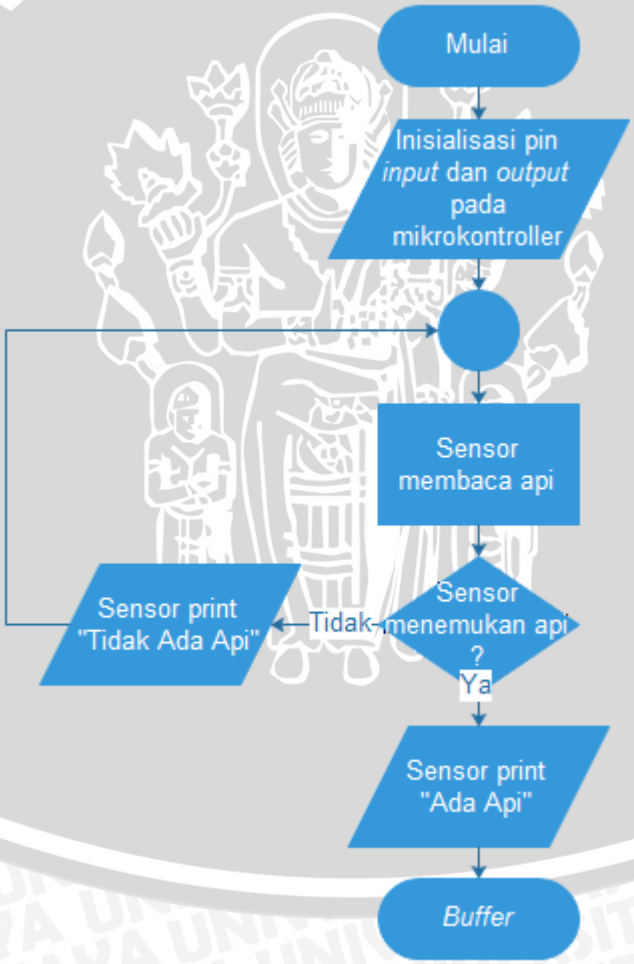
Pada skematik dapat dilihat rangkaian sensor menggunakan *Arduino Nano* sebagai mikrokontroler dan *UVTRON* R9454 sebagai sensor untuk mendeteksi api. Pin *ground UVTRON* dihubungkan ke pin *ground* pada *Arduino*, untuk *inputan* dari *uvtron* masuk ke pin 8 pada *Arduino*, dan untuk *output* untuk serial pada *quadcopter* dan sebagai inputan untuk *buzzer* terdapat pada pin 11 *Arduino.* Karena daya yang dibutuhkan sensor sebesar 12 V maka menggunakan baterai 12 V. Untuk mikrokontroler menggunakan daya dari konektor *USB* 5 V yang terdapat pada sistem *quadcopter*.

5.1.2 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.

5.1.2.1 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 digunkan untuk mendeteksi objek pada sistem rangkaian sensor dapat dilihat pada gambar diagram alir 5.2.

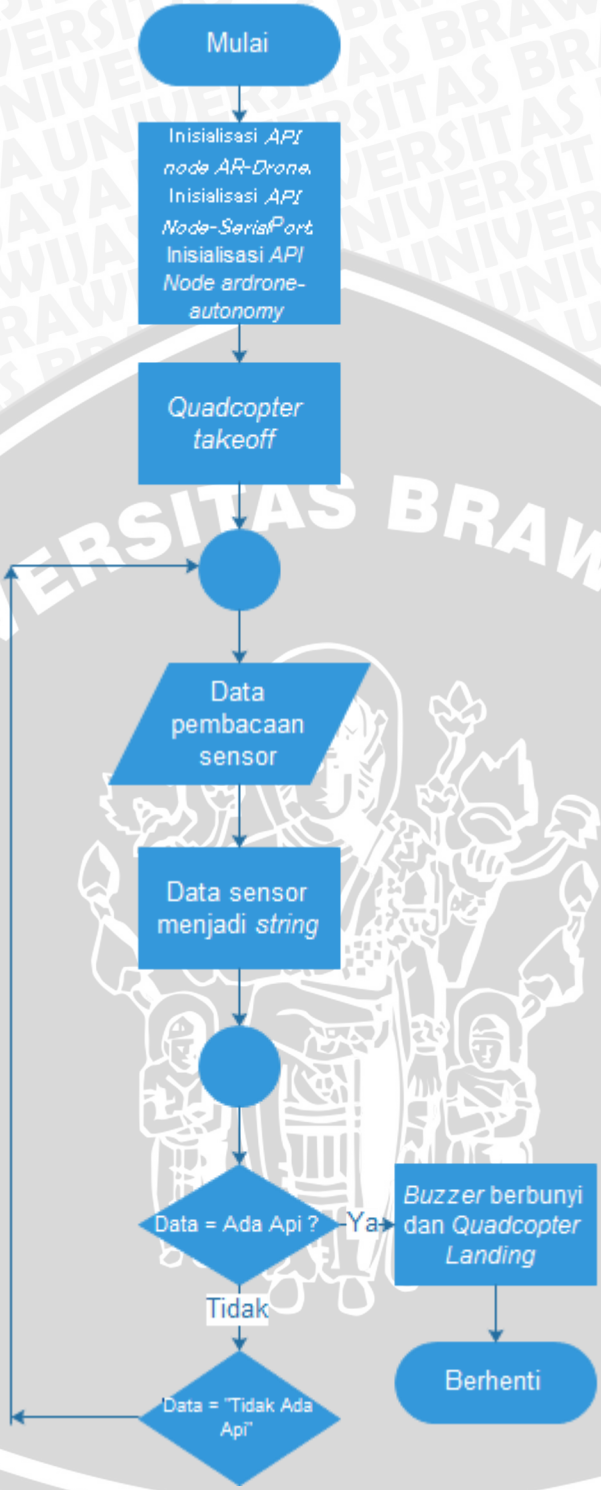


Gambar 5.2 Diagram alir pada rangkaian sensor

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 inisialisasi, maka sensor akan mulai membaca apakah ada sinar api 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 selanjutnya *quadcopter* akan mendarat. Jika api tidak ditemukan, maka pembacaan api terus dilakukan. Rangkaian sensor akan terus mengirimkan hasil pembacaan deteksi api dan dikirmkan secara serial ke *mainboard AR Drone 2.0.*

5.1.2.2 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 5.3.



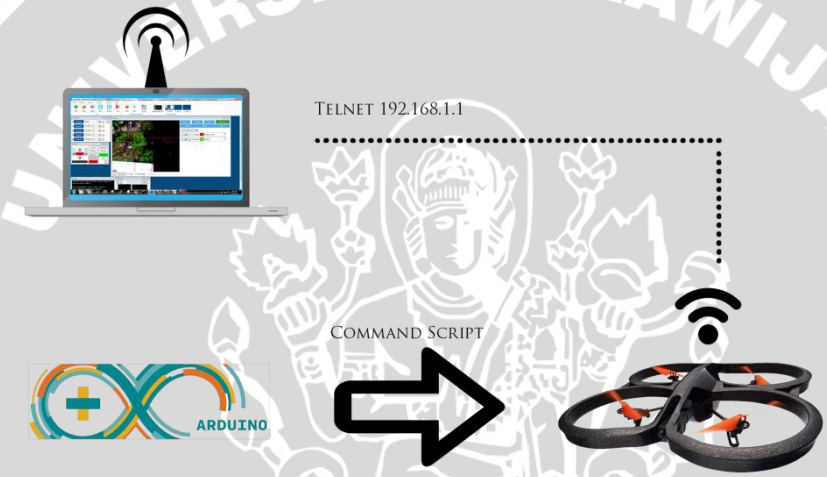
Gambar 5.3 Diagram alir algoritma pada sistem *quadcopter*

Dari gambar diagram alir 5.3 dapat dijelaskan yang pertama kali dilakukan algoritma yang dibangun dalam sistem *quadcopter* adalah penginisialisasian *API Node AR-Drone*, *API Node-SerialPort* dan *Node ardrone-autonomy*. 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 sensor membaca adanya api, maka *quadcopter* akan mendarat dan proses berhenti. Pengecekan kondisi ada tidaknya api selalu dilakukan, jika tidak ditemukan adanya api maka penelusuran dilakukan hingga menemukan api dan memenuhi syarat untuk berhenti.

5.2 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 5.4.



Gambar 5.4 Diagram blok sistem

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.

5.3 Implementasi Sistem

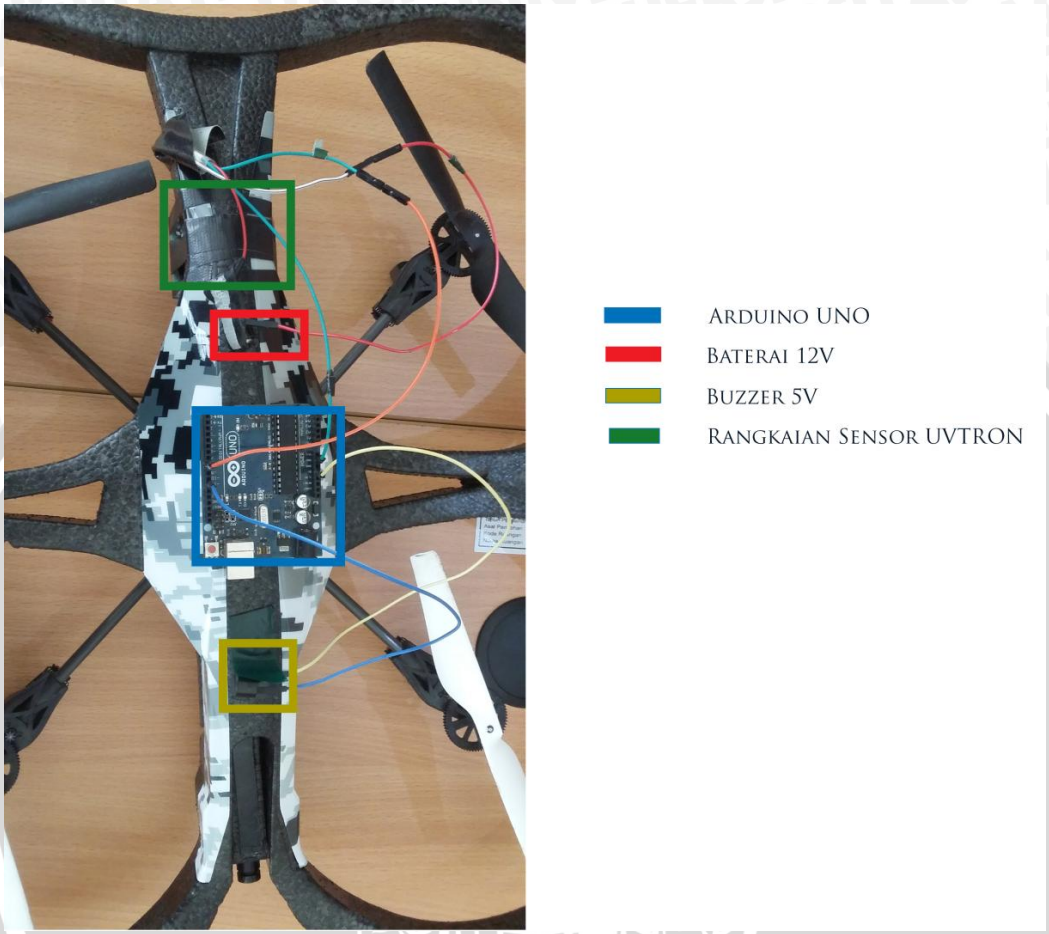
Tahapan implementasi adalah perealisasian dari tahap perancangan sistem yang telah dilakukan sebelumnya. Implementasi sistem harus sesuai dengan perancangan untuk mendapatkan hasil yang sesuai dengan tujuan. Implementasi terbagi menjadi dua bagian yaitu implementasi *hardware* dan implementasi *software*.

5.3.1 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.

5.3.1.1 Rangkaian Sensor

Pada implementasi tahap ini dijelaskan tentang pemasangan sensor *UVTRON* dan juga buzzer gambar dari pengimplementasian rangkaian sensor dapat dilihat pada gambar 5.5.



Gambar 5.5 Implementasi rangkaian sensor

Pada gambar 5.5 implementasi rangkaian sensor digunakan kabel jumper *male-female* untuk menghubungkan setiap komponen menuju pin *arduino.*

Untuk menentukan ketinggian yang ideal dalam pembacaan api dilakukan pengujian sensor didarat dengan jarak 0 sampai dengan 550 cm.

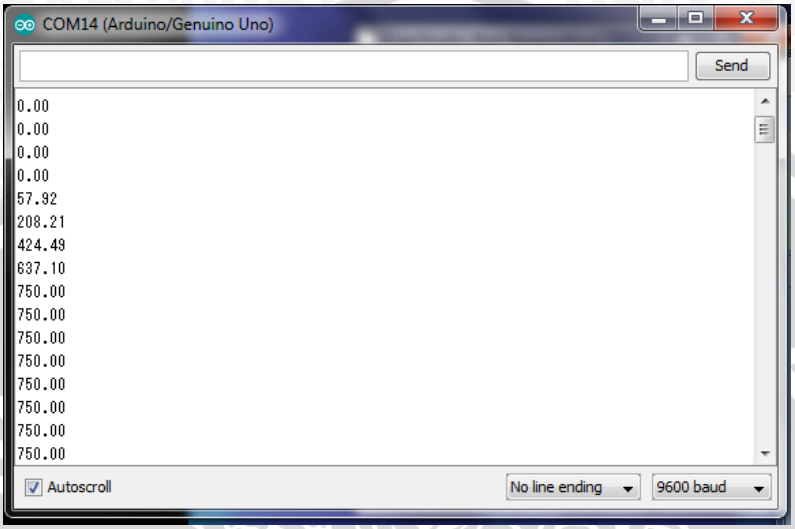
```
1  void setup() {
2   Serial.begin(9600);
3  }
4  void loop() {
5    int sensorValue = analogRead(A0);
6    float voltage = sensorValue * (750.0 / 1023.0);
7    Serial.println(voltage);
8    delay(500);
9  }
```

Gambar 5.6 Kode program pembacaan sensor untuk mengetahui jarak ideal

Penjelasan lebih lanjut dari kode program yang ada pada gambar 5.6 adalah sebagai berikut:

1. Baris 1 sampai baris 3 merupakan inisialisasi *baudrate* yang akan digunakan yaitu 9600.
2. Baris 4 sampai baris 9 merupakan proses perhitungan nilai untuk konversi pembacaan *analog*.

Data hasil pembacaan sensor menggunakan serial monitor *arduino IDE* sebagai berikut:



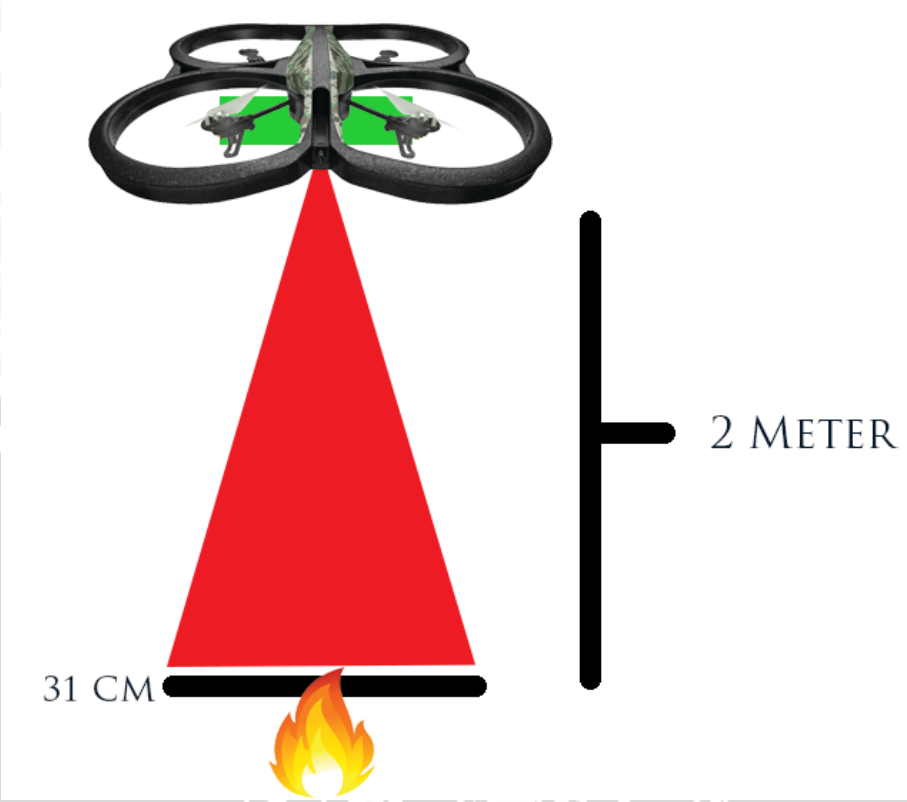
Gambar 5.7 Kode program pembacaan sensor untuk mendeteksi api

Untuk hasil pembacaan yang lebih jelas dikumpulkan dalam tabel yang berisi jarak dan hasil pembacaan sensor pada tabel 5.1.

Tabel 5.1 Tabel hasil pembacaan sensor

No.	Jarak Sensor Dengan Api (cm)	Nilai 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

Pada tabel 5.1 ditunjukkan nilai tertinggi pada jarak 0 cm sampai dengan 200 cm. Berdasarkan percobaan tersebut, ditetapkan ketinggian yang ideal setinggi 200 cm.



Gambar 5.8 Cakupan sensor dan jarak yang ideal dalam mendeteksi api

Pada gambar 5.8 dapat dilihat cakupan sistem dalam mendeteksi api membentuk kerucut dengan ketinggian terbang yang telah ditetapkan setinggi 2 meter.

5.3.1.2 *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 alumunium.

Gambar 5.9 *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 (*Transmiter*)
6. RX (*Receiver*)

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

5.3.2 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.

5.3.2.1 *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 5.10 dapat dilihat letak *Node.JS* pada sistem *quadcopter.*

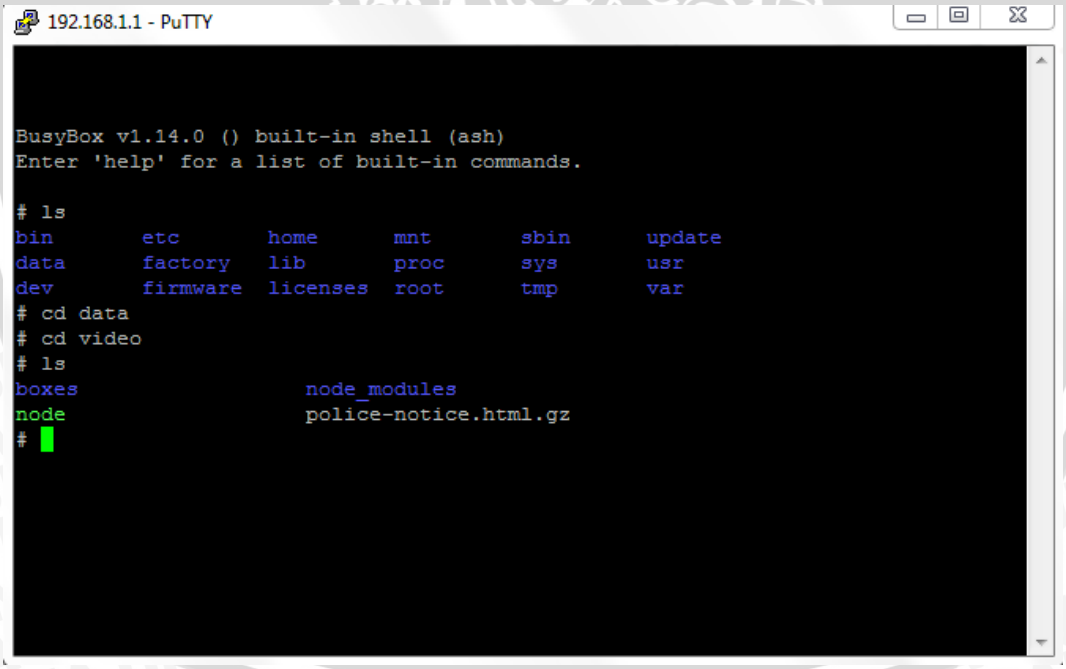
```
BusyBox v1.14.0 () built-in shell (ash)
Enter 'help' for a list of built-in commands.

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

Gambar 5.10 *Node.JS* pada sistem *quadcopter*

5.3.2.2 *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 5.11.



Gambar 5.11 *Node* pada sistem *quadcopter*

BAB 6 PENGUJIAN

Pada bab ini akan menjelaskan terkait dengan pengujian dimulai dari skenario pengujian, proses pengujian serta analisis terhadap data hasil dari pengujian yang dilakukan. Proses pengujian dilakukan untuk mengetahui apakah sistem yang sudah dibuat sesuai dengan analisis kebutuhan yang diingkan. Pengujian terhadap sistem dilakukan dalam beberapa tahapan yang sesuai dengan kebutuhan. Sedangkan analisis dilakukan agar dapat ditarik kesimpulan dari penelitian yang sudah dilakukan.

6.1 Pengujian Deteksi Api menggunakan Rangkaian Sensor

6.1.1 Tujuan Pengujian

Tujuan dilakukan pengujian deteksi api menggunakan rangkaian sensor adalah untuk mengetahui apakah rangkaian sensor yang telah dirancang bekerja sesuai keinginan. Deteksi api merupakan kebutuhan fungsional yang apabila tidak berhasil maka *quadcopter* tidak akan dapat berjalan sesuai seharusnya.

6.1.2 Prosedur Pengujian

Pengujian fungsionalitas sensor dilakukan dengan merangkai sensor *UVTRON* dengan *arduino.* Kemudian mengunggah program deteksi api pada *arduino.* Apabila sensor *UVTRON* mendeteksi api maka *buzzer* dan lampu *led* pada *arduino* akan menyala, jika tidak ditemukan adanya api maka lampu *led* dan *buzzer* akan mati.

6.1.3 Pelaksanaan Pengujian

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.

```
1   #define ledPin 13
2   #define AI 8
3   #define AO 11
4   #define BZ 9
5
6   void setup() {
7     Serial.begin(9600);
8     pinMode(AI,INPUT_PULLUP);
9     pinMode(AO,OUTPUT);
10    pinMode(ledPin, OUTPUT);
11    pinMode(BZ, OUTPUT);
12   }
13
```

Gambar 6.1 Lanjutan

```
14  void loop() {
15  if (digitalRead(AI) == HIGH) {
16    digitalWrite(ledPin, HIGH);
17    Serial.println("A");
18    digitalWrite(AO,HIGH);
19    digitalWrite(BZ, LOW);
20    delay(100);
21  }
22  else
23    {
24      digitalWrite(ledPin, LOW);
25      Serial.println("NA");
26      digitalWrite(AO,LOW);
27      digitalWrite(BZ, HIGH);
28      delay(100);
29    }
30  }
```

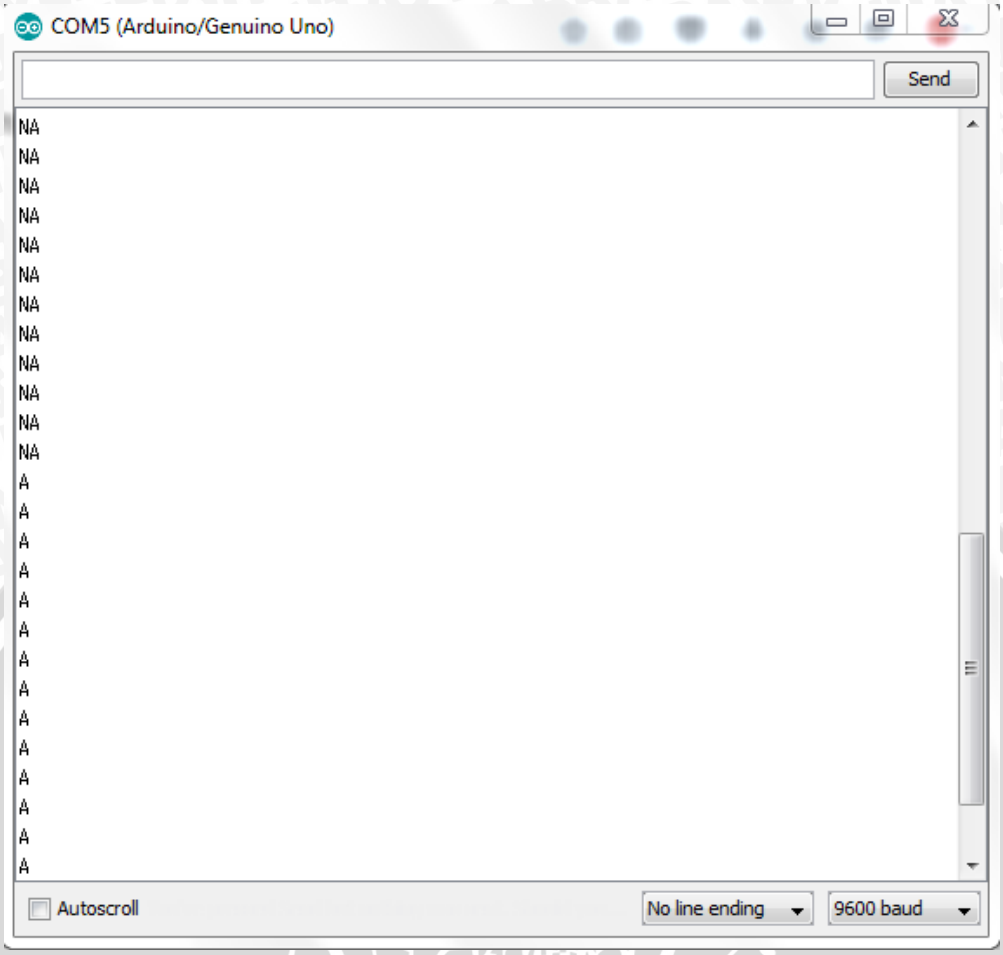
Gambar 6.1 Kode program pembacaan sensor untuk mendeteksi api

Penjelasan lebih lanjut dari kode program yang ada pada gambar 6.1 adalah sebagai berikut:

1. Baris 1 sampai 4 merupakan inisialisasi pin dari *buzzer* dan sensor.
2. Baris 6 sampai 12 merupakan inisialisasi *baudrate*, *input* dan *output* dari setiap pin sensor. Penggunaan *INPUT_PULLUP* ketika sensor bernilai *HIGH* berarti sensor mati, dan bila bernilai *LOW* maka sensor hidup.
3. Baris 14 sampai 30 merupakan baris proses pembacaan api dan apabila hasil pembacaan sensor memenuhi kondisi yang ditetapkan maka kode program akan melakukan *print* dengan tipe data *string* hasil pembacaan sensor.

6.1.4 Hasil Pengujian

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



Gambar 6.2 Tampilan pembacaan sensor pada *Arduino IDE*

6.1.5 Analisis Pengujian

Berdasarkan percobaan pembacaan sensor pada *arduino IDE*, sensor telah dapat membaca adanya api dan tidak adanya api dengan jarak yang ditetapkan pada implementasi rangkaian sensor. Data "A" pada *arduino IDE* berarti ada api, dan data "NA" berarti tidak ada api.

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

6.2.1 Tujuan Pengujian

Tujuan dari pengujian ini untuk mengetahui apakah sensor dapat mengirimkan data hasil pembacaan sensor dan kemudian dilihat berapa lama proses pengiriman data hingga dapat tampil di sistem *quadcopter.*

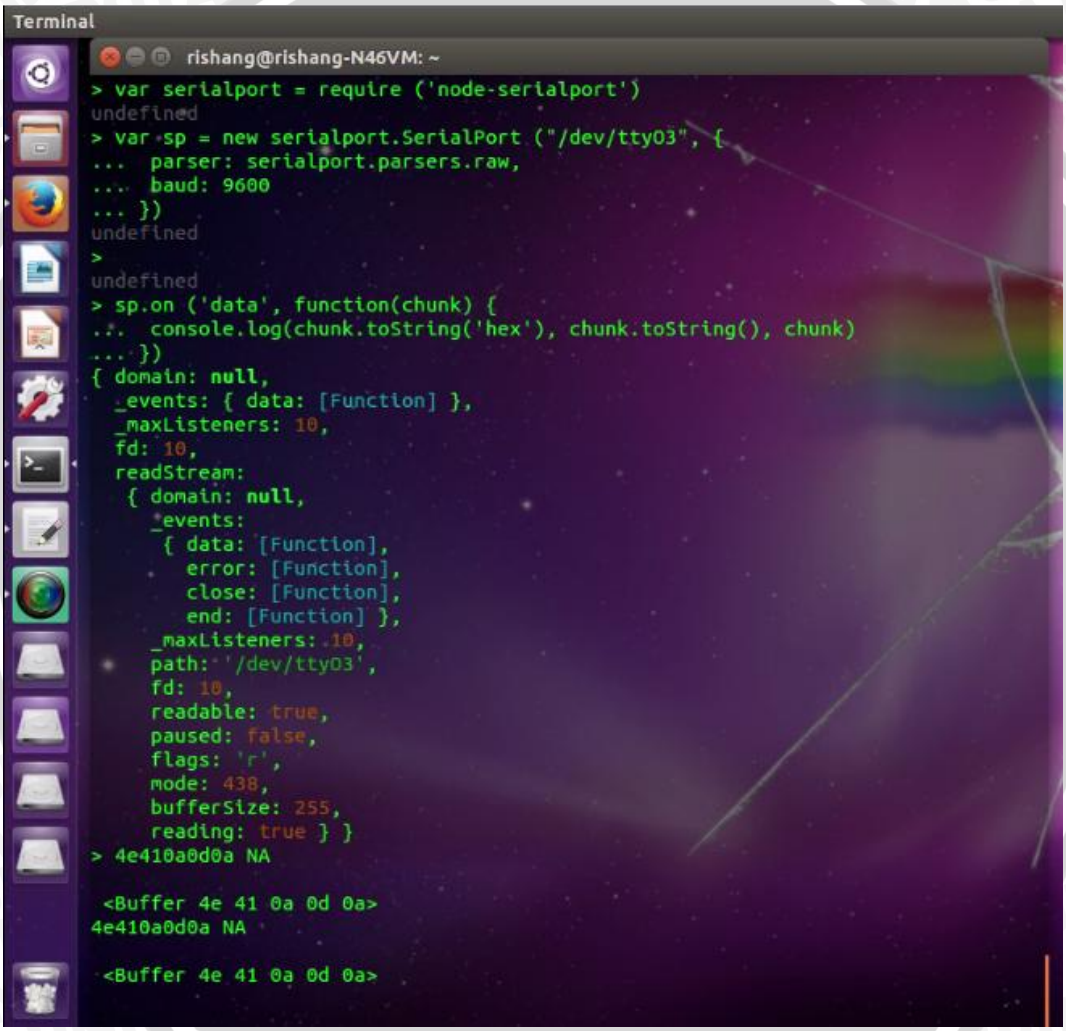
6.2.2 Prosedur Pengujian

Prosedur pengujian pengiriman data melalui jaringan *wireless* dilakukan dengan menguji waktu penerimaan data dari sensor menuju sistem *quadcopter.* Pengujian ini dilakukan saat *quadcopter* didarat. Pengujian ini dilakukan dengan

memberikan titik api selanjutnya sensor di aktifkan dan *quadcopter* diletakkan dari jarak 10 cm sampai 200 cm dari komputer *user*.

6.2.3 Pelaksanaan Pengujian

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. Pada gambar 6.3 merupakan tampilan untuk mengeksekusi *Node.JS*.



Gambar 6.3 Tampilan sistem *quadcopter*

```
1  var serialport = require('node-serialport')
2  var sp = new serialport.SerialPort("/dev/ttyO3", {
3  parser: serialport.parsers.raw,
4  baud: 9600
5  })
6  sp.on('data', function(chunk) {
7  console.log(chunk.toString('hex'), chunk.toString(), chunk)
8  })
```

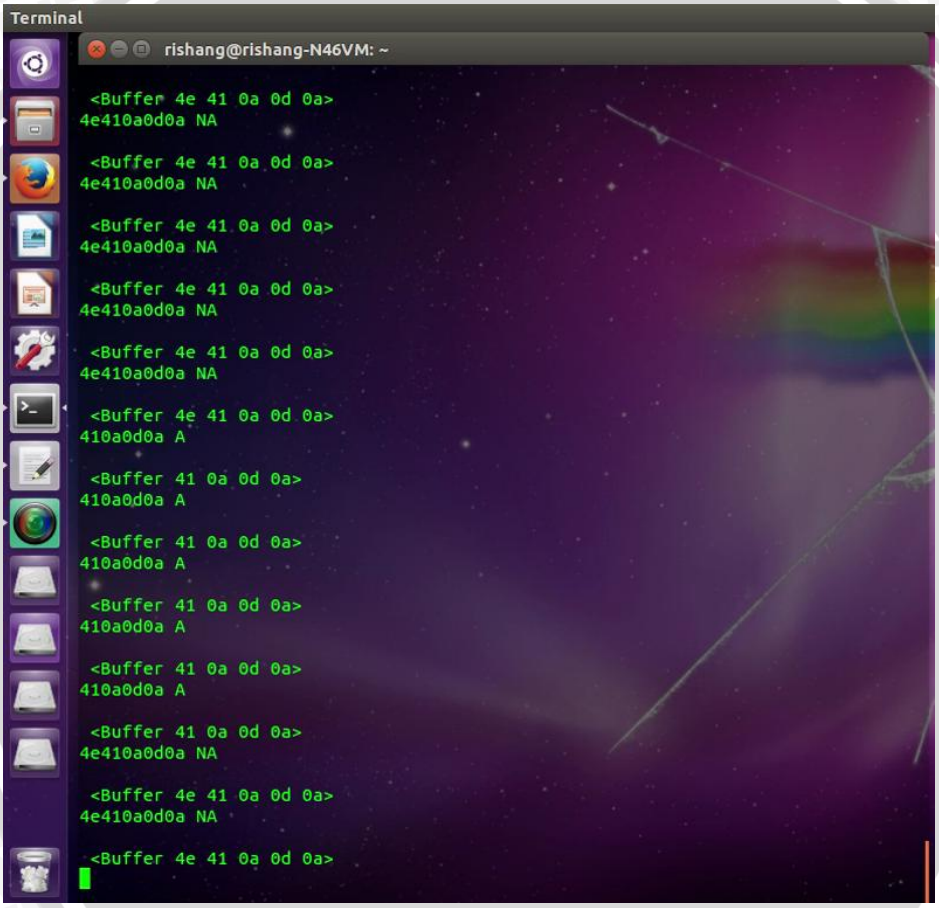
Gambar 6.4 Kode program untuk mengeksekusi *API*

Penjelasan tiap baris dari kode program pada gambar 6.4:

1. Baris 1 merupakan inisialisasi pemanggilan *API serialport.*
2. Baris 2 sampai 5 merupakan inisialisasi *port* yang digunakan. *Mainboard quadcopter* menggunakan *baudrate* 9600 maka pada program akan menyesuaikan. *Serialport* yang digunakan *quadcopter* adalah "ttyO3".
3. Baris 6 sampai 8 merupakan penerimaan data sensor yang akan diubah menjadi data *string* oleh *quadcopter*.

6.2.4 Hasil Pengujian

Dari pengujian didapatkan hasil sebagai berikut:



Gambar 6.5 Tampilan hasil dari pembacaan sensor pada sistem *quadcopter*

6.2.5 Analisis Pengujian

Berdasarkan hasil pengujian yang dilakukan, baik respon waktu pembacaan sensor dan penerimaan data pada sistem *quadcopter* menggunakan *API serialport* memiliki *delay* 100 *milisecon.* Hasil yang didapat ini merupakan hasil yang sama dengan percobaan sebelumnya dilakukan dengan menggunakan *Arduino IDE.* Dari hasil pengujian tersebut dapat dikatakan bahwa respon pengiriman dan pembacaan data sensor pada sistem *quadcopter* menggunakan

API telah dapat menjalankan fungsinya sebagai penghubung sensor dan sistem *quadcopter* yang dibuat.

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

6.3.1 Tujuan Pengujian

Tujuan dari pengujian ini adalah untuk mengetahui kinerja *API Node Ar Drone* yang telah dipasang pada sistem dapat bekerja. Fungsi *API Node Ar Drone* sendiri merupakan salah satu kebutuhan fungsional yang harus terpenuhi oleh sistem.

6.3.2 Prosedur Pengujian

Prosedur pengujian *quadcopter* terbang secara *Autonomous* dengan menguji kesesuaian gerakan *quadcopter* dengan perintah yang telah dirancang. Perintah pengujian berupa perintah *takeoff,* maju, geser ke kanan, mundur, *hover* dan *landing.*

6.3.3 Pelaksanaan Pengujian

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. Kode program terdapat pada gambar 6.6 berikut penjelasannya.

1. Baris 1 dan baris 2 merupakan inisialisasi untuk membuat variabel *mission* dan pemanggilan *library API ardrone-autonomy*.
2. Baris 3 sampai barus 6 merupakan perintah *takeoff* dengan ketinggian 1 meter dan *hover* selama 1000 *milisecond.*
3. Baris 7 sampai dengan 36 merupakan perintah untuk membentuk pola *grid* secara *autonomous. Forward (1)* merupakan gerakan maju sejauh 1 meter, *right (0.3)* merupakan gerakan kanan sejauh 0.3 meter, *left (0.3)* merupakan gerakan ke kiri sejauh 0.3 meter, cw (90) merupakan gerakan searah jarum jam dengan besar sudut 90 derajat, *hover (1000)* merupakan diam stabil ditempat selama 1000 milisecond, *backward (1)* merupakan gerakan mundur sejauh 1 meter, *zero* merukapan perintah untuk kembali ketitik awal ketika *takeoff* dan *land* adalah perintah untuk mendarat.
4. Baris 38 sampai dengan 48 merupakan fungsi untuk menjalankan perintah terbang, tanpa fungsi ini perintah terbang tidak akan berjalan. Fungsi ini juga sebagai penentu situasi jika terjadi *error* atau malfungsi, maka *quadcopter* diperintahkan untuk *landing* secara *emergency* dan

pada terminal akan mencetak "*Oops, something bad happened: %s"* terdapat %s merupakan perintah untuk mencetak tipe data *string.*

```
1  var autonomy = require('ardrone-autonomy');
2  var mission =  autonomy.createMission();
3
4  mission.takeoff()
5  mission.altitude(1)
6  mission.hover(1000)
7  mission.forward(1)
8  mission.hover(1000)
9  mission.right(0.3)
10 mission.hover(1000)
11 mission.backward(1)
12 mission.hover(1000)
13 mission.right(0.3)
14 mission.hover(1000)
15 mission.forward(1)
16 mission.hover(1000)
17 mission.right(0.3)
18 mission.hover(1000)
19 mission.backward(1)
20 mission.hover(1000)
21 mission.left(1)
22 mission.cw(90)
23 mission.left(0.3)
24 mission.hover(1000)
25 mission.forward(1)
26 mission.hover(1000)
27 mission.left(0.3)
28 mission.hover(1000)
29 mission.backward(1)
30 mission.hover(1000)
31 mission.left(0.3)
32 mission.hover(1000)
33 mission.forward(1)
34 mission.hover(1000)
35 mission.zero()
36 mission.land();
37
38 mission.run(function (err, result) {
39     if (err) {
40         console.trace("Oops,   something   bad   happened:   %s",
41 err.message);
42         mission.client().stop();
43         mission.client().land();
44     } else {
45         console.log("Mission success!");
46         process.exit(0);
47     }
48 });
```

Gambar 6.6 Tampilan hasil dari pembacaan sensor pada sistem *quadcopter*

6.3.4 Hasil Pengujian

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 6.1 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 6.2 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 6.3 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 6.4 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%

6.3.5 Analisis Pengujian

Berdasarkan hasil pengujian yang dilakukan, *quadcopter* terbang menggunakan *indoorhul* yang belum dipasangi rangkaian sensor dan telah berhasil menjalankan fungsi terbang secara *autonomous*. Selanjutnya adalah penggabungan fungsi pembacaan api untuk pendeteksian dan mengganti *indoorhul* yang telah dipasangi rangkaian sensor agar *quadcopter* dapat mendeteksi api.

6.4 Pengujian Pencarian Titik Api Menggunakan Metode Pencarian *Grid*

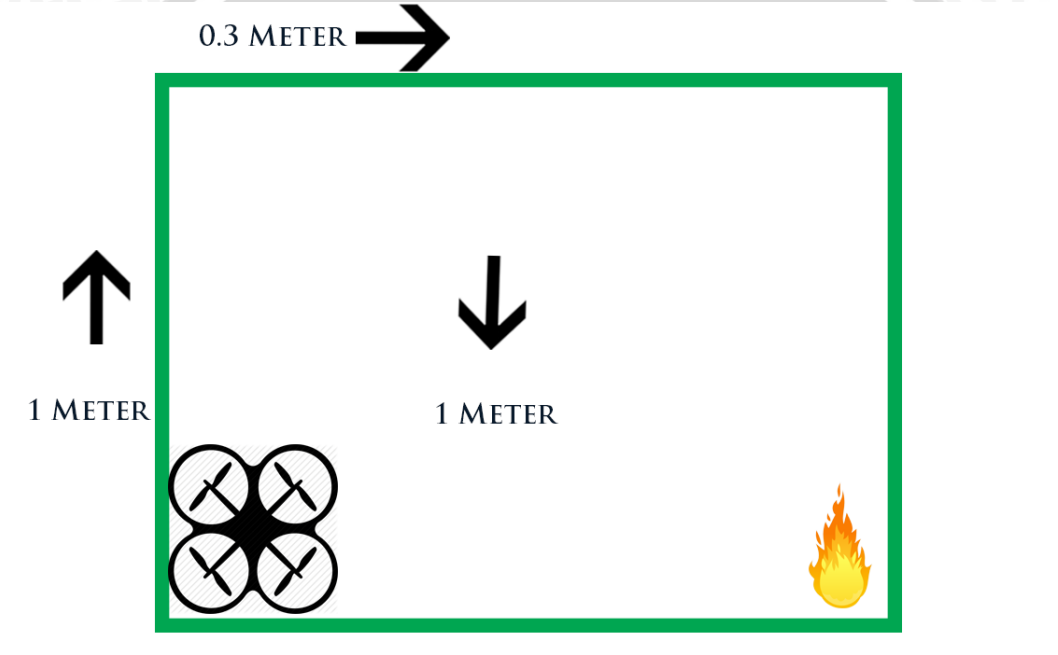
6.4.1 Tujuan Pengujian

Tujuan pengujian ini adalah untuk mengetahui kinerja dan waktu respon *quadcopter* untuk mendeteksi api dengan metode pencarian *grid*. Pengujian ini

merupakan pengujian gabungan fungsionalitas dari pengujian-pengujian sebelumnya.

6.4.2 Prosedur Pengujian

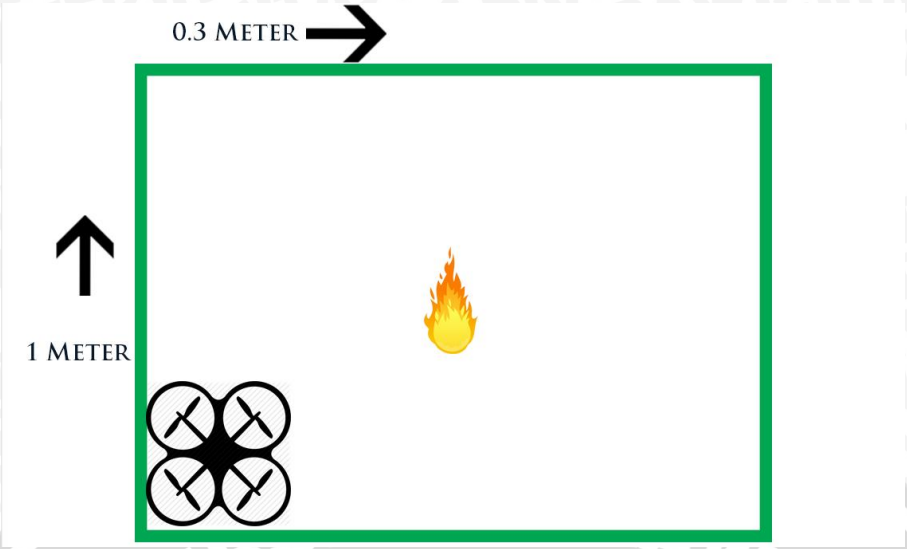
Prosedur pengujian dengan melakukan skenario penerbangan *quadcopter* secara *autonomous* menggunakan kode program yang telah dirancang, kemudian disiapkan sebuah sumber api yang harus dideteksi oleh *quadcopter.* Selanjutnya jika *quadcopter* telah menemukan api, *buzzer* akan berbunyi selama api tersebut menyala, dan *quadcopter* akan *landing* secara *autonomous.* Api ditempatkan pada sudut kanan bawah dari area pengujian. Skenario pengujian lebih jelasnya pada gambar 6.7.



Gambar 6.7 Skenario 1 pengujian pencarian api dengan metode *Grid*

Skenario pengujian pencarian api ini menggunakan metode *grid.* Langkah pencarian api ini adalah quadcopter akan maju sejauh 1 meter, kemudian *quadcopter* akan bergeser ke kanan sejauh 0.3 meter dan mundur sejauh 1 meter, rangkaian pergerakan ini dilakukan sampai 2 kali sehingga *quadcopter* mencapai titik api dan *landing*. Pengujian ini dilakukan berulang kali sampai mendapatkan data yang cukup untuk dianalisa.

Pengujian pencarian api menggunakan metode *grid* ini dilakukan dengan 2 skenario yang berbeda. Skenario pertama api diletakkan pada titik terjauh seperti pada gambar 6.7, yaitu api terdapat pada pojok kanan bawah area pencarian. Skenario kedua api ditempatkan pada titik terdekat seperti gambar 6.8, yaitu api terdapat pada tengah area pencarian. Tiap skenario dilakukan pengambilan data sebanyak 5 kali.



Gambar 6.8 Skenario 2 pengujian pencarian api dengan metode *Grid*

6.4.3 Pelaksanaan Pengujian

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. Kode program terdapat pada gambar 6.10 berikut penjelasannya.

```
1  var autonomy = require('ardrone-autonomy');
2  var mission =  autonomy.createMission();
3  var Api;
4  var serialport = require('node-serialport')
5  var sp = new serialport.SerialPort("/dev/ttyO3", {
6  parser: serialport.parsers.readline("\n"),
7  baud: 9600
8  })
9  sp.on('data', function(chunk) {
10 Api = chunk.toString();
11 console.log("%s",Api);
12
13 mission.takeoff()
14 mission.altitude(1)
15 mission.hover(1000)
16 mission.forward(1)
17 mission.hover(1000)
18 mission.right(0.3)
19 mission.hover(1000)
20 mission.backward(1)
21 mission.hover(1000)
22 mission.right(0.3)
23 mission.hover(1000)
24 mission.forward(1)
25 mission.hover(1000)
26 mission.right(0.3)
27 mission.hover(1000)
28 mission.backward(1)
29 mission.hover(1000)
```

Gambar 6.9 Lanjutan

```
30 if (Api=='A') {
31 console.log("Ada Api!!\n");
32 mission.client().land();
33 }
34 else if (Api=='NA') {
35 console.log("Tidak Ada Api\n");
36 }
37 })
38
39 mission.run(function (err, result) {
40     if (err) {
41         console.trace("Oops,   something   bad   happened:   %s",
42 err.message);
43         mission.client().stop();
44         mission.client().land();
45     } else {
46         console.log("Mission success!");
47         process.exit(0);
48     }
49 });
50
```

Gambar 6.9 Kode proram pengujian sistem

Penjelasan kode program yang ada pada gambar 6.9:

1. Baris 1 sampai baris 3 merupakan inisialisasi *library node ardrone-autonomy* dan inisialisasi variabel yang akan digunakan.
2. Baris 4 sampai baris 11 merupakan inisialisasi *node-serialport* yang digunakan untuk komunikasi serial antara *quadcopter* dan rangkaian sensor.
3. Baris 13 sampai baris 50 merupakan perintah terbang secara *grid* dalam pencarian api.

6.4.4 Hasil Pengujian

Hasil pengujian yang didapatkan dari pelaksanan pengujian pencarian api dengan metode pencarian *grid* skenario 1 dihimpun pada tabel 6.5 dan untuk skenario 2 terdapat pada tabel 6.6. Tabel berisikan kumpulan waktu yang dibutuhkan *quadcopter* untuk mencapai titik api.

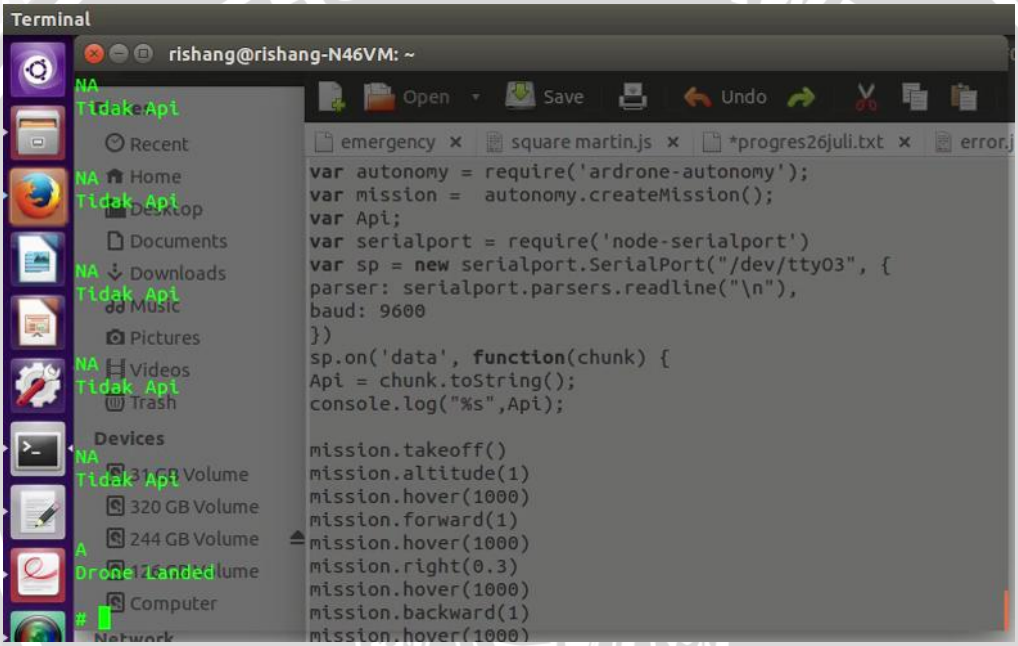
Tabel 6.5 Tabel hasil respon waktu pencarian dan pembacaan api skenario 1

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

Tabel 6.6 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.*



Gambar 6.10 Tampilan *terminal* saat *quadcopter* menemukan api

6.4.5 Analisis Pengujian

Berdasarkan pengujian yang telah dilakukan, didapatkan hasil bahwa pencarian api dengan metode *grid* menempuh waktu 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. Waktu respon akan lebih lama ketika kondisi *quadcopter* tidak stabil, dikarenakan *quadcopter* akan menstabilkan kondisi terbang terlebih dahulu untuk menjalankan perintah selanjutnya. *Quadcopter* yang telah dipasangi rangkaian sensor pada *indoorhull* akan cendrung terbang tidak stabil karena pertambahan bobot *payload,* sehingga terkadang menyebabkan terbang *quadcopter* tidak akan mengikuti perintah yang diberikan. Untuk mengantisipasi hal tersebut perlu ditambahkan perintah *hover* agar *quadcopter* dapat menstabilkan terbangnya.

BAB 7 PENUTUP

Berdasarkan metodologi penelitian yang sudah ditentukan, pada bab ini merupakan tahap terakhir dalam melakukan penelitian setelah melakukan tahap studi literatur, perancangan sistem, implementasi sistem, pengujian serta analisa hasil pengujian dari sistem yang dibangun. Adapun beberapa kesimpulan dan saran yang dapat ditarik adalah sebagai berikut:

7.1 Kesimpulan

1. Sistem ini tediri 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.

7.2 Saran

Beberapa hal yang dapat dikembangkan untuk kesempurnaan sistem ini adalah:

1. Perlu perhitungan lebih lanjut untuk hal berat *payload quadcopter* agar dapat terbang secara normal.
2. Dapat ditambahkan atau digantikan penggunaan sensor dengan citra menggnakan kamera *build-in AR Drone* untuk mendeteksi api, agar *quadcopter* tidak *overweight.*

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_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].

Geisendorfer, Felix., 2013. *Node serialport.* [API]. Tersedia di: <https://github.com/felixge/node-cross-compiler/downloads> [diakses pada 1 April 2016].

Gulotta, Francis., 2014. *Node-serialport*. [API]. Tersedia di: < https://github.com/EmergingTechnologyAdvisors/node-serialport#windows> [Diakses pada 10 Juni 2016].

Jim´enez Lugo, Jacobo dan Zell, Andreas., 2013. *Framework for Autonomous On-board Navigation with the AR.Drone*. University of Tübingen. Tersedia di <http://www.ra.cs.uni-tuebingen.de/publikationen/2013/jimenezJINT2013.pdf> [Diakses pada 6 Mei 2016].

Luukkonen, Teppo., 2011 *"Modelling and control of quadcopter"* Aalto University, Espoo.

Node Package Manager., 2015. NPM. [online] (Terakhir diperbaharui 29 Desember 2015). Tersedia di: <https://npmjs.com/getting-started> [Diakses pada 1 April 2016]

Pracoyo, Agus., 2012. Analisis dan Desain *Library* Penginderaan Sensor Api *Hamamatsu UVTRON* R2868. Jurnal ELTEK, Vol 10 No 02, Oktober 2012 ISSN 1693-4024 [Diakses pada 10 Juni 2016].