DESAIN DAN IMPLEMENTASI PATH FOLLOWING DRONE

Fernaldy Julius[1], **Alfa Satya Putra**[2], **Hendra Tjahyadi**[3]
[1,2,3]Program Studi Sistem Komputer, Fakultas Ilmu Komputer
Universitas Pelita Harapan, Tangerang
e-mail: [1]fernaldy@yahoo.com, [2]alfa.putra@uph.edu, [3]hendra.tjahyadi@uph.edu

ABSTRAK

Pada makalah ini dilakukan perancangan Drone yang dapat bergerak sendiri sesuai jalur yang sudah ditetapkan pada program komputer. Drone yang digunakan adalah Parrot AR. Drone 2.0. Drone memancarkan wifi dan komputer harus terkoneksi dengan wifi Drone untuk mengirimkan data jalur. Bahasa JavaScript dan library ar-drone digunakan untuk pemrograman Drone. Node.js digunakan untuk mengirim program dengan wifi Drone. Pengujian Drone dilakukan menggunakan jalur berbentuk segitiga dengan tiga skenario pengujian, dimana pada masing-masing skenario dilakukan lima kali pengujian. Pada skenario pertama, Drone bergerak di luar ruangan dan mendarat di titik akhir segitiga. Pada skenario kedua, Drone bergerak di luar ruangan dan mendarat di setiap titik segitiga. Pada skenario ketiga, Drone bergerak di dalam ruangan dan mendarat di titik akhir segitiga. Hasil pengujian menunjukkan Drone sudah dapat bergerak mengikuti jalur yang diberikan. Namun karena tidak adanya feedback pada sistem, Drone tidak dapat mendeteksi dan melakukan perbaikan posisi, sehingga mempengaruhi akurasi gerakan Drone.

Kata kunci: drone, javascript, path following

ABSTRACT

This paper discusses the design of an autonomous Drone that can follow a designated path from a computer program. The Drone used is Parrot AR. Drone 2.0. The Drone broadcasts wifi signal and the computer must connect to the Drone wifi in order to send the path data. JavaScript programming language and ar-drone library is used to program the Drone. Node.js is used to send the program using the Drone wifi. Drone testing is done by using a triangular path with three testing scenarios, and each scenario will be tested five times. In the first scenario, Drone moves outdoors and lands only on the finish point of the triangular path. In the second scenario, Drone moves outdoors and lands on each point of the triangular path. In the third scenario, Drone moves indoors and lands only on the finish point of the triangular path. Testing shows that Drone is able to follow the provided path. However, the lack of feedback in the system means the Drone is unable to detect and adjust its position, and this affects the accuracy of Drone's movement.

***Keywords**: drone, javascript, path following*

PENDAHULUAN

Pada zaman teknologi ini sudah tersedia berbagai aplikasi untuk mengantarkan barang yang dipesan atau dikirim ke suatu tempat. Adanya aplikasi tersebut dapat mempermudah orang-orang mengirim barang ataupun membeli tanpa harus datang ke toko.

Saat ini aplikasi tersebut mempunyai jangka waktu pengiriman yang cukup lama untuk sampai ke tempat tujuan yang diinginkan. Hal itu disebabkan karena pengiriman umumnya menggunakan kendaraan beroda yang sangat tergantung dengan kondisi lalu lintas dan kemudahan akses.

Saat ini, *Drone* sudah banyak digunakan untuk memotret dan merekam tempat-tempat yang sulit terjangkau oleh manusia. Selain itu *Drone* juga dapat digunakan sebagai pengganti kendaraan beroda untuk mengirim barang, karena *Drone* bergerak di udara sehingga mempermudah akses dan mempercepat waktu pengiriman.

Untuk membuat *Drone* supaya dapat mengantar barang ke suatu tempat dibutuhkan sebuah perancanaan rute yang disebut *path planning*. *Path planning* berfungsi untuk membuat rute terpendek ke arah tujuan yang sudah ditentukan, sehingga dapat mempercepat waktu untuk sampai ke tujuan. *Drone* dapat diprogram untuk dapat bergerak sendiri (*autopilot*) menelusuri rute atau jalur yang sudah ditentukan.

Salah satu aplikasi dari *Drone* adalah untuk mengirim *blood products*, yang sudah dikembangkan oleh Amukele, Ness, Tobian, Boyd dan Street, dimana blood products dikirim menggunakan *Drone* karena tidak dapat diprediksi kebutuhannya, mempunyai waktu penyimpanan terbatas dan kondisi penyimpanan yang ketat. *Drone* tersebut dikendalikan secara manual ke tempat tujuan oleh seorang pilot di darat. Penelitian tersebut menunjukkan bahwa kondisi *blood product* tidak terpengaruh selama transportasi, sehingga disimpulkan bahwa *Drone* adalah sarana transportasi yang tepat untuk mengirim *blood products* [1].

Portal sudah melakukan penelitian mengenai *Drone* yang bergerak secara *autopilot*, menggunakan sebuah *software* yang dijalankan pada komputer dan dikomunikasikan ke *Drone* melalui jaringan wifi. Java digunakan sebagai bahasa pemrograman pada penelitian tersebut, dan DiaSpec digunakan untuk membuat *framework* pemrograman dan modul terpusat dari sistem *Drone* [2].

Penelitian mengenai *path planning* untuk navigasi Drone secara autopilot sudah dilakukan oleh Li, Zlatanova, Koopman, Bai, dan Diakité, dimana dirancang dua jenis rute untuk navigasi Drone di dalam ruangan, yaitu rute aman yang terpendek (*safe shortest path*, SSP) dan rute aman yang menghemat biaya (*safe least cost path*, SLCP). Keduanya dibuat supaya *Drone* dapat menjaga jarak minimum terhadap rintangan. Disimpulkan bahwa SLCP membuat *Drone* terbang stabil pada ketinggian yang sudah ditetapkan [3].

A. Rumusan Masalah

Dua rumusan masalah yang akan dibahas pada makalah ini adalah apakah *Drone* dapat terbang sesuai jalur yang diinginkan, dan bagaimana cara *Drone* dapat bergerak ke titik yang sudah ditentukan tanpa dikendalikan.

B. Tujuan

Tujuan dari makalah ini adalah mensimulasikan *Drone* sebagai sebuah kurir yang dapat mengantarkan barang tanpa harus dikontrol oleh pengguna. *Drone* akan bergerak mengikuti jalur yang sudah ditentukan secara *offline*.

C. Batasan Masalah

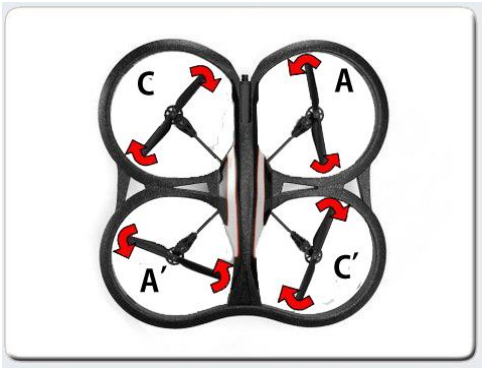
Agar makalah ini tetap sesuai dengan tujuan yang diharapkan, maka ditetapkan beberapa batasan-batasan masalah sebagai berikut:

a. *Drone* bergerak mengikuti jalur yang sudah diberikan.
b. Sistem tidak memiliki *feedback*.
c. Pergerakan *Drone* dikendalikan menggunakan laju (*speed*).

TINJAUAN PUSTAKA

A. Drone

Drone yang digunakan adalah Parrot AR.Drone 2.0 Elite Edition. AR.Drone adalah *quad-rotor* UAS (*Unmanned Aerial Vehicle*) yang menerima perintah melakui koneksi nirkabel. *Drone* dapat diperintah untuk melakukan berbagai jenis aksi seperti *take-off*, *landing*, *hover*, dan beberapa aksi lainnya ketika terbang.



Gambar 1. Rotasi motor

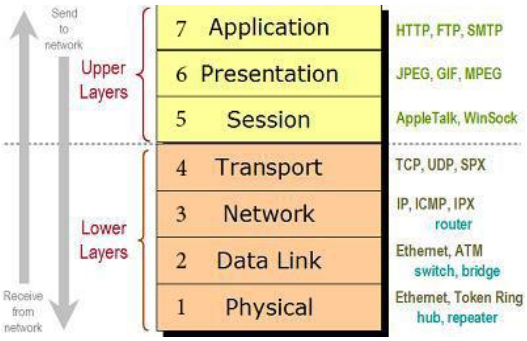
Drone memiliki 2 pasang rotor (Gambar 1). Pasangan pertama adalah A dengan A', yang berputar melawan arah jarum jam. Pasangan kedua adalah C dengan C', yang berputar searah jarum jam.

AR.Drone mempunyai berbagai sensor, seperti akselerometer dan gyrometer untuk mengukur *pitch*, *roll*, dan *yaw*. Pengukuran digunakan untuk memberikan informasi dan menstabilkan *Drone* dengan mengatur nilai *pitch*, *roll*, dan *yaw*. *Ultrasound telemeter* disediakan untuk mengukur ketinggian, menstabilkan ketinggian, dan membantu mengontrol kecepatan vertikal.

Drone memiliki dua buah kamera: kamera depan dan kamera bawah. *Drone* dapat merekam, memotret, dan melihat secara langsung melalui kamera. *Drone* juga memiliki *wireless connection* sendiri, dimana *Drone* berperan sebagai *host*, dan perangkat yang terkoneksi akan berperan sebagai *client* untuk mengendalikan *Drone*. Wifi *Drone* akan menyala otomatis ketika *Drone* menyala [4].

B. OSI Model

OSI *(Open System Interconnection*) Model bertujuan untuk menyediakan seperangkat standar desain untuk peralatan manufaktur sehingga dapat berkomunikasi satu sama lain. OSI Model memiliki 7 lapisan (*layer*) seperti Gambar 2 di bawah.



Gambar 2. OSI Layer

Layer semakin ke atas akan semakin dekat dengan *user*. Sebaliknya, *layer* semakin ke bawah akan semakin dekat dengan mesin [5]. Komunikasi untuk pengendalian AR.Drone dilakukan menggunakan protokol UDP (*User Datagram Protocol*) yang beroperasi di *Layer* 4 atau *Transport Layer*. UDP digunakan untuk mengirim data dari device ke *Drone* dan sebaliknya [4].

C. JavaScript

JavaScript adalah sebuah bahasa *scripting*, umumnya digunakan di web.

JavaScript digunakan untuk menambahkan fitur pada HTML (*Hypertext Markup Language*) dan biasanya ditemukan di dalam kode HTML. JavaScript adalah bahasa yang ditafsirkan, sehingga tidak perlu dikompilasi. JavaScript me-*render* halaman web secara interaktif dan dinamis, sehingga membuat halaman web dapat bereaksi terhadap *event*, menampilkan efek khusus, menerima variabel berupa teks, memvalidasi data, membuat *cookies*, dan mendeteksi *web browser* pengguna [6].

D. Node.js

Node.js adalah *runtime* lingkungan JavaScript yang bekerja secara asinkron. Node.js didesain untuk membangun aplikasi jaringan yang dapat dikembangkan secara terukur. Karena proses dijalankan secara asinkron, Node.js tidak melakukan penguncian (*blocking*) yang dapat terjadi karena menunggu operasi diluar JavaScript selesai dieksekusi, sehingga tidak akan terjadi *dead-lock* pada Node.js [7].

E. Library ar-drone

Library ar-drone adalah *library* dari JavaScript yang digunakan untuk pengkodean AR.Drone. *Library* ini menyediakan fungsi-fungsi yang dapat digunakan untuk memprogram *Drone* secara mudah. *Library* ar-drone dapat di-install menggunakan Node.js *command prompt* [8]. Fungsi-fungsi yang dipakai dari *library* adalah sebagai berikut:

1. client.takeoff(callback);

   Digunakan untuk *Drone* lepas landas.
2. client.land(callback);

   Digunakan untuk *Drone* mendarat.
3. client.up(speed);

   Digunakan untuk meningkatkan ketinggian *Drone*, *speed* yang digunakan berkisar dari 0 hingga 1, dari kecepatan yang sudah diatur di dalam aplikasi *Drone*.
4. client.clockwise(speed) / client.counterClockwise(speed);

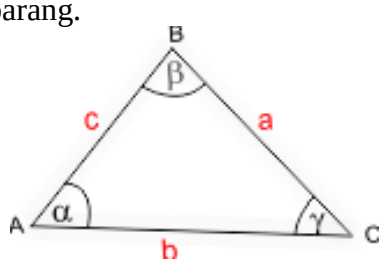
   Digunakan untuk membuat *Drone* berputar ke arah jarum jam (kanan) atau ke arah berlawanan jarum jam (kiri), speed yang digunakan berkisar dari 0 hingga 1, dari kecepatan yang sudah diatur di dalam aplikasi *Drone*.
5. client.front(speed);

   Digunakan untuk *Drone* bergerak maju, *speed* yang digunakan berkisar dari 0 hingga 1 dari kecepatan yang sudah diatur di dalam aplikasi *Drone*.
6. client.stop();

   Digunakan untuk membuat semua gerakan *Drone* menjadi 0.

F. Trigonometri Aturan Sinus

Kata trigonometri berasal dari dua suku kata bahasa Yunani, yaitu *trigonon* yang artinya tiga sudut, dan *metro* yang artinya mengukur [9]. Trigonometri adalah ilmu matematika yang mempelajari tentang sudut, sisi, dan perbandingan antara sudut terhadap sisi. Trigonometri memiliki tiga aturan, yaitu: sinus, kosinus, dan luas segitiga. Melalui aturan sinus, fungsi geometri sinus dapat digunakan dalam segitiga sembarang.



Gambar 3. Segitiga Sembarang

Aturan sinus menyatakan bahwa dalam sebuah segitiga sembarang seperti pada Gambar 3, perbandingan panjang sisi dan sinus dari sudut yang

berhadapan dengan sisi tersebut mempunyai nilai sama untuk semua pasangan sisi dan sudut yang berhadapan. Aturan tersebut dinyatakan dalam persamaan (1) di bawah.

$$\frac{a}{\sin \alpha} = \frac{b}{\sin \beta} = \frac{c}{\sin \gamma} \quad (1)$$

G. *Speed, Distance, Time, Rotation, Rotation Speed*

Laju (*speed*) sebuah benda dapat dihitung dengan persamaan (2), yaitu jarak (*distance*) dari satu tempat ke tempat lain dibagi waktu (*time*) yang dibutuhkan untuk berpindah dari satu tempat ke tempat yang lain.

$$S = {}^{d}/_{t} \quad (2)$$

Dengan S = *speed* (m/detik), d = jarak atau *distance* (m), dan t = waktu atau *time* (detik). Menggunakan konsep yang sama, laju putaran (*rotation speed*) sebuah benda dapat dihitung dengan persamaan (3), yaitu sudut (*angle*) dibagi waktu (*time*) yang dibutuhkan untuk berputar.

$$\omega_{deg} = {}^{\theta}/_{T} \quad (3)$$

Dengan ω_{deg}= *rotation speed* (°/detik), θ = *angle* (°), dan T = *time* (detik) [10].

H. *Roll, Pitch, Yaw*

Gambar 4. *Roll, Pitch, Yaw*

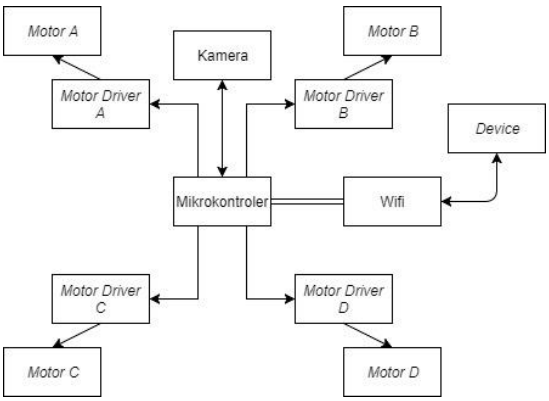
Gerakan *Drone* dapat dibagi tiga yaitu, *pitch, roll,* dan *yaw* seperti pada Gambar 4. *Roll* berguna untuk bergerak miring ke arah sisi kanan *(+roll)* atau sisi kiri *(-roll). Pitch* berguna untuk bergerak maju *(+pitch)* atau mundur *(-pitch)* 5. *Yaw* berguna untuk berotasi ke kiri *(+yaw)* atau ke kanan*(-yaw)* [11].

PERANCANGAN

Drone digunakan sebagai prototipe untuk membantu memindahkan barang atau sebagai kurir. Perbedaan dari kurir-kurir yang saat ini sudah ada yaitu *Drone* lebih cepat, dikarenakan *Drone* terbang di udara.

Pada sistem prototipe ini, *Drone* akan berjalan ke dua titik tujuan dan kembali lagi ke tempat semula. *Drone* bergerak secara *offline* sesuai perintah yang diberikan. *Drone* memiliki *wifi* sendiri, dan perintah akan diberikan dari device melalui wifi *Drone* menggunakan Node.js *Command Prompt*.

Pengkodean menggunakan *library* ar-drone yang di *download* melalui Node.js *Command Prompt*. JavaScript memanggil *library* ar-drone untuk melakukan fungsi-fungsi yang terdapat di dalam *library* tersebut. Kode akan di kirim ke *Drone* melalui *wifi* Drone menggunakan Node.js *Command Prompt*. *Drone* akan berjalan sesuai dengan instruksi yang sudah dibuat dalam *coding*.

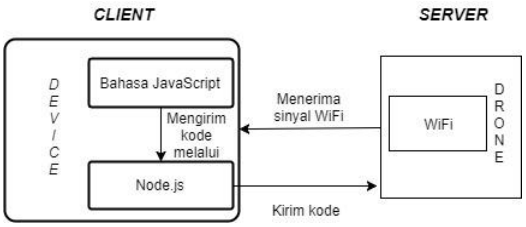


Gambar 5. Diagram Blok AR.Drone

Pada diagram blok AR.Drone di Gambar 5, mikrokontroler *Drone* dapat

mengatur laju putaran 4 motor secara bersamaan dan mampu membuat motor *Drone* seimbang ketika *Throttle*, *Hover*, *Pitch*, *Roll*, dan *Yaw*. Mikrokontroler menggerakkan motor melalui *motor driver* yang berfungsi untuk menggerakkan motor dengan mengambil sinyal kontrol arus rendah dan mengubahnya menjadi sinyal arus yang lebih tinggi yang bisa menggerakkan motor.

Wifi Drone digunakan untuk menerima perintah yang diberikan dari *device* ke mikrokontroler untuk menjalankan perintah yang diberikan oleh *device*. Kamera menangkap gambar dan video, lalu dikirim ke mikrokontroler. Mikrontroler mengirim data ke *device* dan menampilkan *streaming* video yang ditangkap dari kamera.



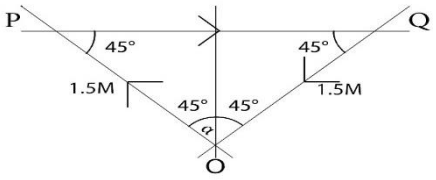
Gambar 6. Diagram Blok Sistem

Pada diagram blok sistem di Gambar 6, *wifi Drone* akan memancarkan sinyal dan diterima oleh *device*. Pengkodean menggunakan bahasa JavaScript yang akan dikirim melalui Node.js *command prompt*. Node.js membantu mengirimkan kode melalui *wifi* ke *Drone*.

PENGUJIAN DAN ANALISIS

A. Plotting

Plotting yang dipakai untuk pengujian sistem *path following Drone* adalah *plotting* seperti pada Gambar 7.



Gambar 7. *Plotting*

Plotting untuk jalur atau *path* yang digunakan berbentuk segitiga, untuk mensimulasikan pengiriman barang ke dua tempat yang berbeda lalu kembali ke tempat semula.

Sisi depan *Drone* adalah sisi yang mempunyai kamera depan. *Drone* akan berada di titik O dan akan bergerak ke arah titik P dengan berputar berlawanan arah jarum jam atau *counter-clockwise* sebesar 45° dan bergerak maju sejarak 1.5 meter. Jarak titik P ke Q menggunakan persamaan (4) seperti di bawah.

$$\sin \propto = \frac{\frac{1}{2}PQ}{OP} \qquad (4)$$

Dari persamaan (4) akan didapatkan jarak P ke Q, sebagai berikut:

1. $\sin 45° = \frac{\frac{1}{2}PQ}{1.5\,M}$
2. $\frac{1}{2}\sqrt{2} = \frac{\frac{1}{2}PQ}{1.5\,M}$
3. $\frac{1}{2}\sqrt{2} \times 1.5\,M = \frac{1}{2}PQ$
4. $PQ = \frac{1}{2}\sqrt{2} \times 1.5\,M \times 2$
5. $PQ = 2.12\,M$

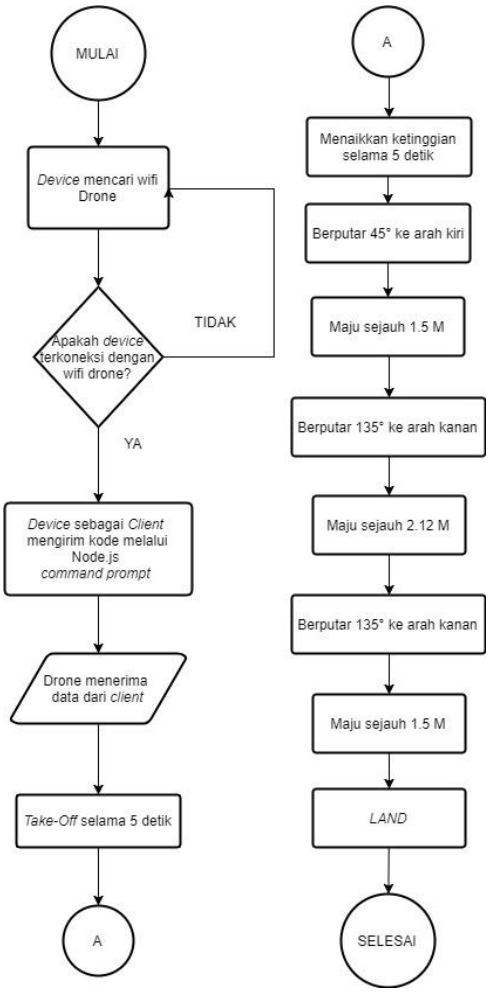
Drone dari titik P akan berputar 135° (180° - 45°) searah jarum jam atau *clockwise* menuju titik Q, dan berjalan maju sejauh 2.12 meter. Sampai di titik Q, *Drone* berputar 135° (180° - 45°) searah jarum jam atau *clockwise* menuju titik O, dan berjalan maju sejauh 1.5 meter.

Sistem ini akan diuji menggunakan tiga jenis skenario pengujian, dimana pada masing-masing skenario akan dilakukan lima kali

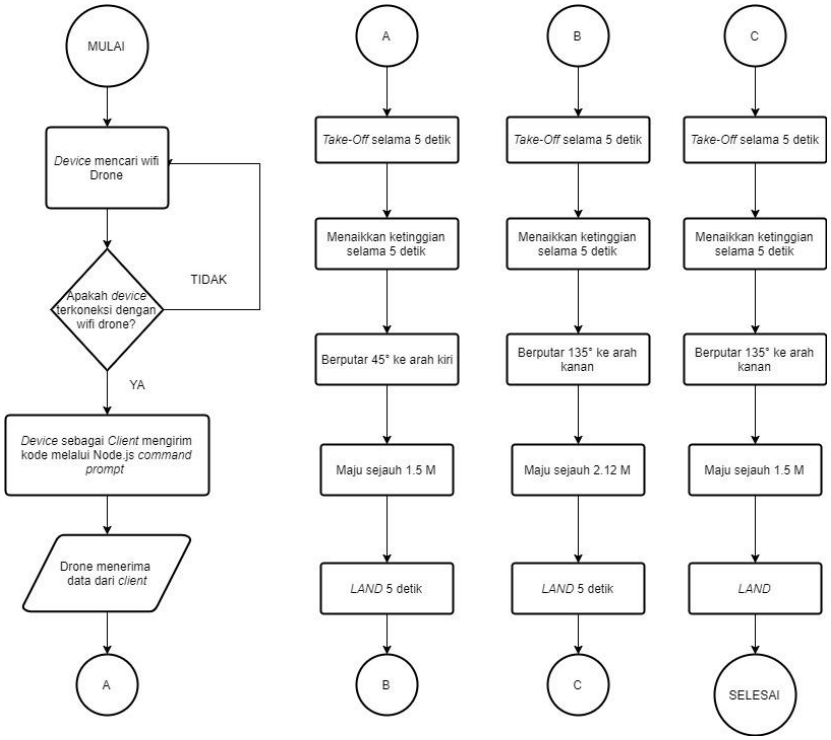
percobaan. Skenario pertama disebut Plot A dimana *Drone* akan dijalankan di luar ruangan dan *Drone* bergerak mengikuti jalur pada Gambar 7 tanpa mendarat. Skenario kedua disebut Plot B, dimana *Drone* akan dijalankan di luar ruangan dan bergerak mengikuti jalur pada Gambar 7 dengan mendarat di tiga titik yang membentuk segitiga. Skenario ketiga disebut Plot C, dimana *Drone* akan dijalankan di dalam ruangan dan bergerak mengikuti jalur pada Gambar 7 tanpa mendarat.

Gambar 8 adalah diagram alir atau *flowchart* untuk Plot A dan Plot C. Keduanya memiliki pergerakan yang sama, namun perbedaan *Plot A* dan *C* adalah *Plot A* dijalankan di luar ruangan *(outdoor)* dan *Plot C* dijalankan di dalam ruangan *(indoor)*. *Drone* akan bergerak dari titik O ke P, dari P ke Q, dan dari Q kembali lagi ke O seperti pada Gambar 7 tanpa mendarat (*land*). *Device* (dalam percobaan ini digunakan laptop) akan mencari *wifi* yang dipancarkan oleh *Drone*. Jika *device* tidak terkoneksi dengan *wifi Drone*, maka *device* harus mengulang mencari *wifi* tersebut hingga terkoneksi. Jika sudah terkoneksi dengan *wifi Drone*, *device* akan berperan sebagai *client* untuk mengirim kode program melalui Node.js *command prompt* menggunakan *wifi*.

Drone akan menerima kode melalui *wifi*, dan bergerak sesuai dengan kode yang sudah diberikan. Setelah menerima kode tersebut, *Drone* akan *take-off* dari titik O selama 5 detik. Setelah 5 detik, *Drone* akan menaikkan ketinggian selama 5 detik. Setelah 5 detik menaikkan ketinggian, *Drone* akan berputar 45° ke arah kiri. Setelah berputar, *Drone* akan maju ke arah titik A sejarak 1.5 meter, lalu berputar 135° ke arah kanan. Setelah berputar, *Drone* akan maju ke titik B sejarak 2.12 meter, lalu akan berputar lagi 135° ke arah kanan. Setelah berputar, *Drone* akan maju ke arah titik O sejarak 1.5 meter, lalu *Drone* akan langsung mendarat (*land*).



Gambar 8. *Flowchart* Plot A dan C



Gambar 9. Flowchart Plot B

Pada Gambar 9, *Drone* akan bergerak di luar ruangan (*outdoor*) dari titik O ke P, lalu dari P ke Q, dan terakhir dari Q kembali ke O seperti pada Gambar 7, dimana *Drone* akan mendarat (*land*) pada ketiga titik O,P dan Q. Awalnya sama seperti Plot A, *device* akan mencari *wifi Drone*. Jika belum terkoneksi, maka *device* harus mengulang mencari *wifi* tersebut hingga terkoneksi. Jika sudah terkoneksi dengan *wifi Drone*, *device* akan berperan sebagai *client* untuk mengirim kode program melalui Node.js *command prompt* menggunakan *wifi Drone*. *Drone* akan menerima data melalui *wifi*, dan akan bergerak sesuai dengan kode yang sudah diberikan.

Setelah menerima kode tersebut, *Drone* akan *take-off* dari titik O selama 5 detik. Setelah itu, *Drone* akan menaikkan ketinggian selama 5 detik, dan berputar 45° ke arah kiri. Setelah berputar, *Drone* akan maju ke titik P sejauh 1.5 meter, lalu *Drone* akan mendarat (*land*) di titik P. *Drone* akan mulai *take-off* kembali dari titik P setelah 5 detik, lalu menaikkan ketinggian selama 5 detik dan berputar 135° ke arah kanan. Setelah berputar, *Drone* akan maju ke arah titik Q sejauh 2.12 meter, lalu mendarat (*land*) di titik Q.

Drone akan mulai *take-off* kembali dari titik Q setelah 5 detik, lalu menaikkan ketinggian selama 5 detik dan berputar 135° ke arah kanan. Setelah berputar, *Drone* akan maju ke arah titik O sejauh 1.5 meter, lalu *Drone* akan mendarat (*land*) di titik O.

B. Estimasi Waktu Pergerakan *Drone*

Bagian ini menjelaskan tentang estimasi waktu yang dibutuhkan *Drone* untuk berotasi dan bergerak maju. Laju maksimal untuk berotasi dari *Drone* adalah 200°/s, dan laju maksimal untuk bergerak maju adalah 5m/s. Laju rotasi yang ditentukan adalah 0.2 kali dari laju

maksimal berotasi, dan laju untuk bergerak maju yang ditentukan adalah 0.25 kali dari laju maksimal bergerak maju. Data waktu akan digunakan di dalam pengkodean pergerakan *Drone*. Penghitungan estimasi waktu pergerakan *Drone* dapat dilihat pada persamaan (5) hingga (8), dimana penghitungan menggunakan persamaan (3) untuk berotasi dan persamaan (2) untuk bergerak maju.

Rotasi 45°:
- 45°=200°/s×0.2×T
- T=45°÷40°/s
- T=1.125 s=1125 ms (5)

Rotasi 135°:
- 135°=200°/s×0.2×T
- T=135°÷40°/s
- T=3.375 s=3375 ms (6)

Bergerak Maju 1.5 Meter:
- 1.5m=5m/s×0.25 ×T
- T=1.5m÷1.25m/s
- T=1.2 s=1200 ms (7)

Bergerak Maju 2.12 Meter:
- 2.12m=5m/s×0.25×T
- T=2.12m÷1.25m/s
- T= 1.696 s=1696 ms (8)

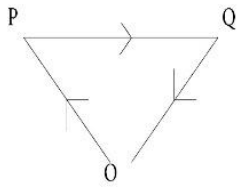
C. Pembahasan Hasil

Tabel 1. Perbandingan Desain Dengan Percobaan

Desain	Hasil Percobaan
Drone dapat terbang membentuk segitiga sesuai plot yang sudah didesain	Drone dapat terbang membentuk segitiga, tetapi tidak sesuai dengan plot
Dapat menaikkan ketinggian yang cukup tinggi	Ketinggian kurang
Jarak yang ditempuh sesuai dengan plot	Jarak yang ditempuh ada yang tidak sesuai dengan plot
Kecepatan sesuai dengan perhitungan	Kecepatan pada saat dilapangan berbeda dengan perhitungan

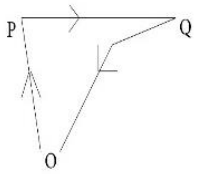
Hasil percobaan dapat dilihat pada: https://youtu.be/DW1euWqWn9s. Perbandingan antara desain dengan hasil percobaan secara keseluruhan dapat dilihat pada Tabel 1. Pada saat bergerak maju, *Drone* bergerak tidak sesuai dengan jarak yang diinginkan, maka waktu untuk mencapai jarak yang diinginkan dinaikkan dua kali lipat. Kecepatan *Drone* bergerak maju di *indoor* lebih cepat dari *outdoor*.

Untuk skenario pengujian plot A (Gambar 10), *Drone* dari titik O ke P berputar sekitar 45° dan bergerak sekitar 1.5 meter. Dari titik P ke Q, *Drone* berputar sekitar 135° dan bergerak sekitar 3 meter yang seharusnya 2.12 meter. Dari titik Q ke O, *Drone* berputar sekitar 135° dan bergerak sekitar 1.5 meter.



Gambar 10. Hasil Plot A

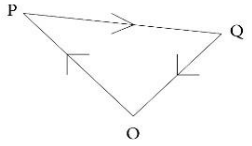
Untuk plot B (Gambar 11), dari titik O ke P *Drone* berputar kurang dari 45°, dan bergerak sekitar 1.5 meter. Dari titik P ke Q, *Drone* berputar sekitar 135° dan bergerak sekitar 3 meter. Dari titik Q ke O, terjadi ketidakseimbangan *Drone* ketika *take-off* dan berputar, sehingga *Drone* terbang ke arah kanan dan bergerak maju kurang dari 1.5 meter.



Gambar 11. Hasil Plot B

Untuk plot C (Gambar 12), dari titik O ke P *Drone* berputar sekitar 45°

dan bergerak sekitar 1.5 meter. Dari titik P ke Q, *Drone* berputar lebih dari 135° dan bergerak sejauh sekitar 2.5 meter. Dari Q ke O, *Drone* bergerak kurang dari 1.5 meter.



Gambar 12. Hasil Plot C

KESIMPULAN

Drone sudah dapat terbang dan mengikuti jalur yang diberikan dengan cukup akurat dari ketiga skenario pengujian yang sudah dilakukan. Faktor angin, pengaturan *propeller* pada *Drone*, dan tingkat daya baterai sangat berpengaruh terhadap akurasi gerak *Drone*. Sistem yang dirancang tidak memiliki *feedback*, sehingga tidak dapat mendeteksi posisi *Drone* dan melakukan perbaikan posisi apabila diperlukan.

Untuk pengembangan selanjutnya dapat dilakukan pemberian *feedback* untuk membantu menangani pengaruh angin atau faktor lain. Salah satunya adalah menggunakan *state* dan *event* untuk membantu memberikan kondisi ketika *Drone* menyimpang dari jalur yang diharapkan. Selain itu dapat diterapkan pengendalian setiap motor *Drone* untuk membantu *Drone* bergerak sesuai jalur. Selain itu juga dapat ditambahkan fitur untuk mengirim *input* dari *Drone* ke komputer yang berisi informasi ketinggian dan kecepatan *Drone* ketika maju atau berputar, supaya pergerakan *Drone* dapat dihitung menggunakan *acceleration.*

DAFTAR PUSTAKA

[1] Amukele, T., Ness, P.M., Tobian, A.A.R., Boyd, J. & Street, J. 2016. Drone Transportation of Blood Products. Transfusion, (Online), 57(3), (https://www.researchgate.net/publication/310474300, diakses 5 Mei 2019).

[2] Portal, J.V. 2011. A Java Autopilot for Parrot A.R. Drone Designed with DiaSpec, (Online), (https://lume.ufrgs.br/handle/10183/37168, diakses 13 Mei 2019).

[3] Li, F., Zlatanova, S., Koopman, M., Bai, X., Diakité, A. 2018. Universal path planning for an indoor drone. Automation in Construction, Volume 95: 275-283

[4] Piskorski, S., Brulez, N., Eline, P., D'Haeyer, F. 2012. AR.Drone Developer Guide, (Online), (https://developer.parrot.com/docs/SDK2/ARDrone_SDK_2_0_1.zip, diakses 13 Mei 2019)

[5] Tanenbaum, A. S., Wetherall, D.J. 2010. Computer Networks: Fifth Edition. Boston: Pearson.

[6] Flanagan, D. 2011. *JavaScript: The Definitive Guide: Sixth Edition*. Sebastopol: O'Reilly Media.

[7] Dahl, R. 2009. Node.js, (Online), (https://nodejs.org/en/, diakses 13 Mei 2019)

[8] Geisendörfer, F. 2012. A node.js client for controlling Parrot AR Drone 2.0 quad-copter, (Online), (https://github.com/felixge/node-ar-drone, diakses 13 Mei 2019)

[9] Boyer, C.B., Merzbach, U.C. 2011. A History of Mathematics, 3rd Edition. San Fransisco: Jossey-Bass.

[10] Kanginan, M. 2008. *Seribu Pena Fisika SMA/MA Kelas X*. Jakarta: Erlangga

[11] Hong, H. 2012. Lesson 3: Flight Movements - Robolink, (Online), (https://www.robolink.com/snap-lesson-3/. diakses 14 Mei 2019)