

BAB 4 ANALISIS KEBUTUHAN

4.1 Gambaran Umum Sistem

Pada gambar 4.1 merupakan gambaran sistem yang akan bekerja, dimana komputer *host* dihubungkan dengan *quadcopter* melalui jaringan *wi-fi* dan *quadcopter* berfungsi sebagai *Server*-nya. Komputer *host* berkomunikasi dengan sistem *quadcopter* melalui ROS (*Robot Operating System*). Logika *fuzzy* dijalankan melalui *node* pada ROS (*Robot Operating System*).



Gambar 4.1 Diagram blok perancangan sistem

Pada sistem kendali *take-off quadcopter AR.Drone* ini terdapat *input* dari sistem yaitu berupa data sensor dan juga navigasi dari *quadcopter AR.Drone* yang nantinya data tersebut akan di jadikan sebagai *input* dari logika *fuzzy*. Sedangkan *output* dari sistem ini adalah *quadcopter* dapat mengatur kecepatan motor dengan *rule* yang telah dibuat. Pemograman yang digunakan untuk penelitian ini adalah menggunakan *fuzzy* editor yang telah disediakan oleh Matlab sebagai program untuk menyusun logika *fuzzy*. Sedangkan untuk pengambilan dan pengiriman data dari komputer menuju *quadcopter* menggunakan bahasa pemograman Python dan di jalankan pada ROS (*Robot Operating System*).

4.2 Kebutuhan Pengguna

Pada bagian ini pengguna dapat berinteraksi dengan sistem menggunakan terminal yang terdapat pada *Robot Operating System* (ROS) dan *gazebo* simulator. Pengguna dapat melihat data sensor, navigasi, dan juga data kecepatan motor saat melakukan proses *take-off*.

4.3 Kebutuhan Sistem

Pada kebutuhan sistem terdiri dari tiga tahapan analisis kebutuhan yaitu berupa kebutuhan perangkat keras, kebutuhan perangkat lunak, serta kebutuhan komunikasi.

4.3.1 Kebutuhan Perangkat Keras

Terdapat beberapa kebutuhan perangkat keras yang digunakan pada penelitian antara lain sebagai berikut;

1. *Laptop* sebagai *Ground station*

Laptop dengan sistem operasi Linux Ubuntu 14.04 pada Gambar 4.2 digunakan sebagai perangkat untuk mengirim perintah navigasi pada *quadcopter* dan digunakan untuk penulisan program yang nantinya akan di implementasikan pada *quadcopter*. Sedangkan *quadcopter* dapat berkomunikasi dengan *ground station* atau *laptop* yaitu dengan menggunakan jaringan *wi-fi*.



Gambar 4.2 Laptop Sebagai *Ground station*

2. *Quadcopter AR-Drone 2.0*

Jenis *quadcopter* yang digunakan dalam penelitian ini adalah *Parrot Ar Drone 2.0* seperti yang terlihat pada Gambar 4.3.



Gambar 4.3 Parrot Ar Drone 2.0

Pada *Parrot Ar Drone 2.0* terdapat beberapa perangkat yang digunakan pada penelitian ini yaitu;

a. *Mainboard AR Drone 2.0*

Mainboard AR Drone 2.0 adalah sebuah mikrokomputer yang berfungsi sebagai pusat pemroses data, baik untuk menghubungkan antara *user* dan *quadcopter*, dan sebagai pengendali komponen yang terdapat pada *quadcopter*, *mainboard* ini menggunakan sistem operasi *linux* yang dapat diakses dengan *telnet* melalui *PC-Desktop*.

b. Sensor Ultrasonik

Sensor ultrasonik digunakan untuk menghitung jarak antara *ground* dan *quadcopter*, sensor ini telah terpasang pada *Parrot AR Drone 2.0* yang berfungsi untuk mengontrol ketinggian pada *quadcopter*.

c. Sensor Inertial Measurement Unit (IMU)

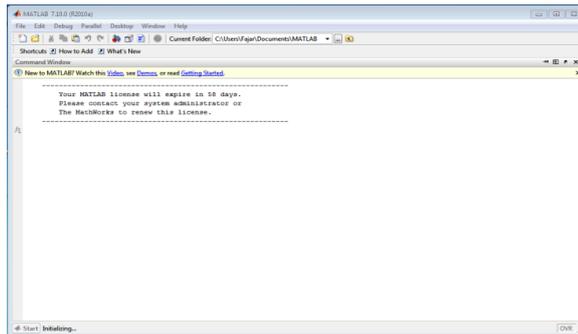
Sensor IMU merupakan sensor yang memanfaatkan sistem pengukuran seperti *gyroskop* dan akselerometer untuk memperkirakan posisi relatif, kecepatan, dan akselerasi dari gerakan motor. sensor ini telah terpasang pada *Parrot AR Drone 2.0* data dari sensor IMU yaitu data percepatan dan juga data perubahan percepatan yang didapat dari akselerometer inilah yang nantinya data tersebut akan di olah menjadi input dari logika *fuzzy*.

4.3.2 Kebutuhan Perangkat Lunak

Terdapat beberapa kebutuhan perangkat lunak yang digunakan pada penelitian antara lain;

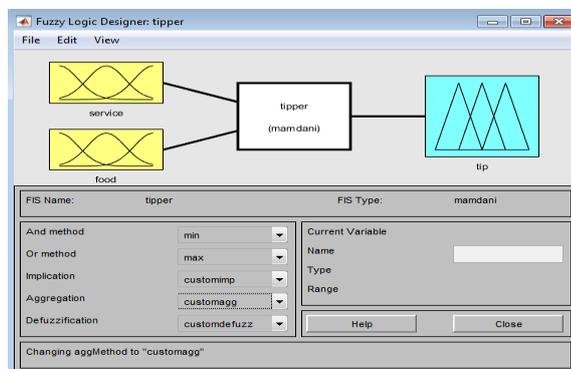
1. Matlab

Pada penelitian ini Matlab digunakan sebagai aplikasi pembuatan *system fuzzy logic*, karena pada matlab disediakan *fuzzy logic toolbox*. *Fuzzy logic toolbox* memberikan fasilitas *graphical user interface (GUI)* untuk mempermudah dalam membangun suatu sistem *fuzzy*. Gambar 4.4 merupakan gambaran dari aplikasi Matlab.



Gambar 4.4 Tampilan Matlab

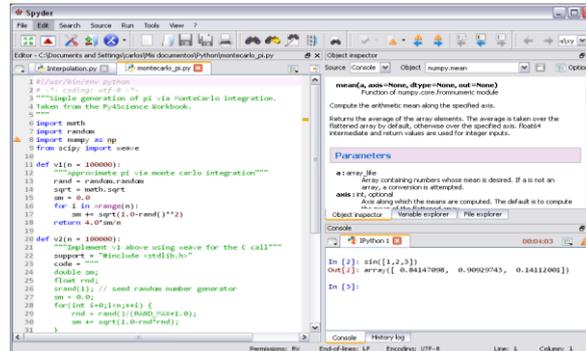
Gambar 4.5 adalah gambaran dari *fuzzy inference system (FIS)* editor yang terdapat pada *fuzzy logic toolbox*. Dengan menggunakan dari *fuzzy inference system (FIS)* editor kita dapat membaca dan memodifikasi data tentang sistem yang akan kita bangun.



Gambar 4.5 Tampilan *fuzzy inference system*

2. Spyder

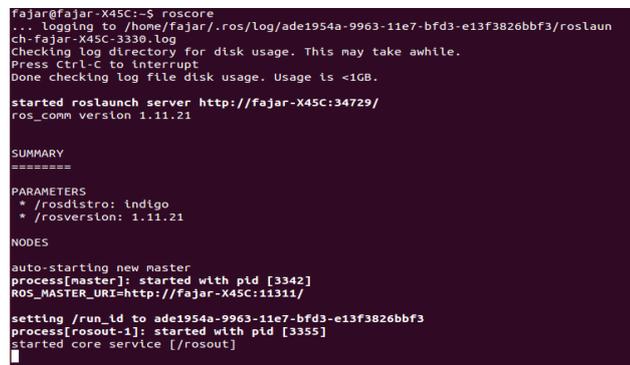
Pada penelitian ini bahasa pemrograman yang digunakan adalah bahasa pemrograman *Python*, sedangkan *Spyder* berfungsi sebagai *Interface Development Environment (IDE)* dari bahasa pemrograman *python*. Program yang telah dibuat nantinya akan dirubah menjadi *node* dan akan di jalankan pada *quadcopter* menggunakan bantuan *Robot Operating System*. Gambar 4.6 merupakan gambar tampilan awal dari *Spyder*.



Gambar 4.6 Spyder sebagai Python IDE

3. Robot Operating System (ROS)

Robot Operating System (ROS) berfungsi sebagai sistem operasi yang akan menjalankan *quadcopter* dengan cara menghubungkan ROS dengan *quadcopter* melalui koneksi *Wi-fi*. *ROS* memiliki peran yang sangat penting pada penelitian ini karena *ROS* merupakan *framework* yang digunakan untuk menghubungkan antara *user* dan *quadcopter*. Dengan menggunakan aplikasi ini, *user* dapat mengakses seluruh data navigasi dari *quadcopter* secara *realtime*. *ROS* sendiri tidak memiliki *Graphic User Interface (GUI)* seperti sistem operasi lainnya. Untuk mengoperasikan sistem operasi ini *user* hanya menggunakan *terminal* untuk mengoperasikannya. Gambar 4.7 merupakan gambar tampilan dari *ROS*.



Gambar 4.7 Robot Operating System (ROS)

4. Gazebo

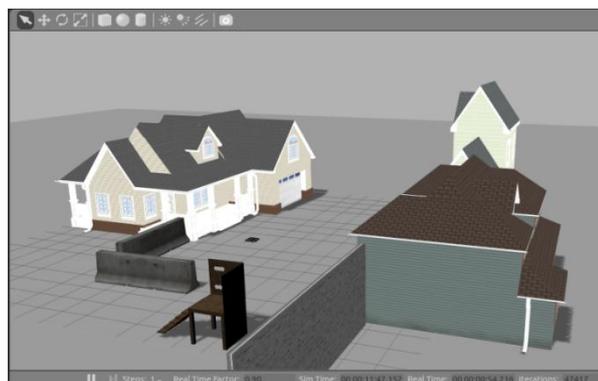
Gazebo merupakan perangkat lunak yang digunakan sebagai *simulator* dari penelitian yang akan dibuat. Pemilihan simulator *gazebo* sendiri bertujuan untuk meminimalisir resiko kerusakan pada *quadcopter* saat menguji algoritma yang telah dibuat. Gambar 4.9 merupakan tampilan awal dari *gazebo*.



Gambar 4.9 Software simulator gazebo

5. Tum Simulator

Tum simulator adalah perangkat lunak yang digunakan sebagai *simulator quadcopter Parrot AR Drone 2.0*. Tum simulator memiliki kesamaan dengan *quadcopter* sesungguhnya, sehingga untuk pengimplementasian pada *quadcopter* sesungguhnya tidak merubah program yang telah dibuat. Gambar 4.9 merupakan tampilan dari tum simulator.



Gambar 4.9 Tampilan tumsimulator

4.4 Kebutuhan Fungsional

Kebutuhan yang harus dipenuhi pada penelitian ini adalah kebutuhan fungsional, karena kebutuhan ini adalah bagaimana sistem yang dibangun harus sesuai dengan sistem yang diinginkan, adapun kebutuhan fungsional yang dibutuhkan adalah sebagai berikut:

1. *Quadcopter* dapat terbang secara *otomatis*

Quadcopter terbang dengan pergerakan vertikal setinggi 1 sampai dengan 4 meter dengan program yang di *remote* ke *quadcopter* menggunakan *Robot Operating System* (ROS), dan program yang digunakan adalah *Python*.

2. Sensor *ultrasonik* membaca jarak antara *ground* dengan *quadcopter*.

Jarak antara *ground* dengan *quadcopter* dapat dibaca oleh sensor *ultrasonik* yang sudah tertanam pada *quadcopter* secara *realtime*. Nantinya hasil pembacaan data sensor ini digunakan untuk proses *take-off*.

3. Sensor IMU berfungsi sebagai pengatur posisi dari *quadcopter*.

Posisi sudut *x,y,z* dapat dibaca oleh sensor IMU yang telah tertanam pada *quadcopter* secara *realtime*. Nantinya hasil pembacaan data sensor ini digunakan untuk proses *take-off*.

4. *Quadcopter* dapat *take-off* menggunakan sistem kendali logika *fuzzy*

Sistem kendali *take-off* menggunakan logika *fuzzy* dapat mempertahankan posisi *hover* pada ketinggian tertentu sesuai *rule* yang telah diimplementasikan pada *quadcopter*.

4.5 Kebutuhan Non-Fungsional

Pada kebutuhan non-fungsional akan dijelaskan mengenai apa saja yang menjadi batasan dalam kebutuhan perancangan sistem. Adapun kebutuhan non-fungsional yang dibutuhkan adalah sebagai berikut.

4.5.1 Karakteristik Pengguna

Dalam penelitian ini *user* melakukan implementasikan program melalui *Robot Operating System* dengan mengubah terlebih dahulu program tersebut menjadi sebuah node lalu dijalankan pada ROS. *User* bertugas sebagai pilot untuk memberikan perintah pada *quadcopter* agar terbang dengan gerakan vertikal untuk berbagai jenis ketinggian. Selain berperan sebagai pilot *user* juga bertugas untuk mendapatkan data navigasi dari *quadcopter*. Data yang telah diperoleh nantinya dijadikan sebagai acuan untuk menentukan kinerja dari sistem yang telah di uji.

4.5.2 Lingkungan Operasi

Pada penelitian ini terdapat beberapa persyaratan lingkungan operasi untuk menjalankan penelitian ini yaitu diantaranya adalah:

1. Landasan yang digunakan pada penelitian ini harus rata, agar pembacaan jarak oleh sensor *ultrasonik* yang ada pada *quadcopter* dapat melakukan pembacaan jarak dengan tepat.
2. Pengujian dilakukan pada ruangan yang tinggi dan luas selain itu ruangan harus tertutup hal ini bertujuan untuk mengurangi *noise* atau gangguan yang disebabkan oleh angin dan tekanan udara.

4.5.3 Asumsi dan Ketergantungan

Terdapat beberapa asumsi ketergantungan pada penelitian ini yaitu:

1. Pada penelitian ini sistem hanya dapat di implementasikan pada *quadcopter* berjenis *Parrot AR Drone 2.0*.
2. Sistem akan berfungsi ketika komputer telah terhubung dengan *wi-fi* yang dipancarkan *quadcopter*.

4.5.4 Batasan Perancangan dan Implementasi

Adapun tujuan dari batasan perancangan dan implementasi ini adalah untuk penentu batasan sistem agar penelitian ini menjadi lebih terarah dan berjalan sesuai harapan yaitu:

1. Pada sistem yang dibangun menggunakan jarak 1 sampai dengan 4 meter dengan selisih jarak 1 meter untuk setiap pengujian
2. Input yang digunakan pada penelitian ini adalah nilai dari percepatan dan perubahan percepatan
3. Pada pengujian akurasi lebih di tekankan pada tingkat akurasi *AR-drone* terhadap ketepatan posisi pengujian dan waktu yang dibutuhkan sesuai dengan jarak yang di uji.