

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Pada sub bab ini akan membahas langkah-langkah penelitian terlebih dahulu yang dilakukan dalam studi ini. Perencanaan penelitian dimulai dengan merumuskan pertanyaan penelitian yang relevan dan signifikan. Melalui tinjauan literatur yang mendalam, studi ini mengidentifikasi kerangka teoritis yang relevan dan penelitian terdahulu yang telah dilakukan dalam bidang yang sama.

Pada Penelitian yang dilakukan oleh [2] yang berjudul “Rancang Bangun Sistem *Monitoring* Muatan Balon Atmosfer pada Kompetisi Muatan Balon Atmosfer Tahun 2019 Lembaga Penerbangan dan Antariksa Nasional (Lapan) Kabupaten Garut “ bertujuan untuk membuat suatu rancangan sistem *monitoring* yang dapat digunakan untuk menampilkan dan menyimpan data profil vertikal atmosfer secara *real-time* bagi pengguna. Sistem *monitoring* muatan balon atmosfer mampu mendeteksi suhu, kelembaban, tekanan udara, ketinggian dalam bentuk grafik. Beberapa komponen *hardware* yang digunakan dalam penelitian ini yaitu telemetri 3DR 433MHz sebagai radio, Sensor GPS UBlox Neo-7m (Sensor GPS), dan BME 280 (sensor suhu, kelembaban, tekanan). Lalu untuk komponen softwarena memakai Visual Studio 2019 dengan bahasa pemrograman C#.

Pada penelitian yang dilakukan oleh [6] yang berjudul “Rancang Bangun *Payload* dan *Interface Monitoring* Roket Dalam Visualisasi 3D dan Pengambilan Foto” bertujuan untuk menganalisa kinerja dari payload dan juga pengambilan foto saat di udara beserta interfacenya. Beberapa komponen *hardware* yang digunakan dalam penelitian ini yaitu Xbee Pro sebagai radio, Sensor GY801, Kamera VC0706. Lalu untuk komponen softwarena memakai *Visual Studio* 2019 dengan bahasa pemrograman C#. Hasil perangkat lunak yang digunakan adalah untuk melakukan pengamatan perilaku roket dan menampilkan hasil foto.

Pada penelitian yang dilakukan oleh [7] yang berjudul “Sistem Telemetry Pada *Ground Control Station* (GCS) Muatan Balon Atmosfer (*Payload*) Menggunakan Delphi 7” membahas mengenai perancangan dan realisasi sistem telemetry menggunakan aplikasi Delphi7 yang digunakan untuk menampilkan sebuah parameter data dari alat ukur muatan balon atmosfer yang dikerjakan oleh peneliti lain. Data yang diperoleh dari hasil pengujian disimpan ke dalam file berformat .txt. Beberapa parameter yang terdapat pada penelitian ini yaitu *pressure*, *temperature*, *humidity*, *latitude*, *longitude*, dan *altitude*.

Pada penelitian yang dilakukan oleh [8] yang berjudul “Pembangunan Aplikasi *Ground Control Station* pada *Unmanned Aerial Vehicle* Berbasis *Internet of Things*” bertujuan untuk memudahkan operator atau pengawas memantau kondisi UAV dari jarak jauh dan mengevaluasi hasil pengujian terbang dari UAV. Beberapa komponen *hardware* yang digunakan yaitu BMP180, HMC5883L (sensor kompas), Neo6MV2 dan MPU6050. Penelitian ini menggunakan web sebagai antarmuka ke pengguna.

Pada penelitian yang dilakukan oleh [9] yang berjudul “Pengembangan Aplikasi *Ground Control Station* (GCS) Untuk Pengawasan dan Pengendalian UAV” bertujuan untuk mengembangkan aplikasi *ground control station* yang mampu memudahkan pengguna dalam mengendalikan dan memantau kondisi UAV di udara secara *real-time* dengan menampilkan data keadaan dan parameter penerbangan UAC sesuai dengan kebutuhan pengguna. Beberapa komponen *hardware* yang digunakan seperti *Flight Controller* 3DR Pixhawk Mini, *Telemetry* 433 MHz, dan GPS. Lalu untuk *software* atau penghubung antara laptop dengan *flight controller* sendiri menggunakan IDE Visual Studio.

2.2 Sistem Monitoring

Sistem monitoring merupakan suatu proses untuk mengumpulkan data dari berbagai sumber daya. Biasanya data yang dikumpulkan merupakan data yang *real-time*. Aksi yang terjadi antara proses-proses dalam sebuah sistem monitoring yaitu bentuk *service*, dimana suatu proses terus menerus berjalan pada interval waktu

tertentu. Proses-proses yang terjadi pada suatu sistem *monitoring* dimulai dari pengumpulan data seperti *network traffic*, *hardware information*, dan lain-lain yang kemudian data tersebut akan dianalisis pada proses analisis data dan pada akhirnya data tersebut akan di tampilkan [10].

Sistem merupakan sekelompok elemen yang terkait erat yang bekerja sama untuk mencapai tujuan tertentu. *Monitoring* didefinisikan sebagai suatu siklus kegiatan yang meliputi pengumpulan, peninjauan, pelaporan dan tindakan atas informasi suatu proses yang sedang diimplementasikan. Berdasarkan uraian diatas tentang sistem dan *monitoring*, dapat disimpulkan bahwa sistem *monitoring* adalah sistem pemantauan yang merupakan rangkaian dari siklus kegiatan yang meliputi pengumpulan data, pelaporan, dan tindakan atas informasi suatu proses yang sedang diimplementasikan [2].

2.3 Ground Control Station

Ground Control Station merupakan sistem *monitoring* yang dapat memvisualisasikan gerak ataupun sikap UAV saat terbang, data yang didapatkan dari sensor UAV akan diterima oleh sistem monitoring dan diolah ke dalam bentuk visual gerak ataupun sikap UAV pada saat terbang [8].

Ground Control Station atau yang bisa disebut juga sebagai stasiun kontrol bumi merupakan stasiun terminal darat yang dibangun sebagai komunikasi antara pesawat ruang angkasa dengan pusat kontrol di bumi ataupun penerimaan gelombang radio dari sumber radio pesawat luar angkasa (astronomi). *Aerial robot* sangat membutuhkan GCS, hal ini disebabkan karena GCS mempunyai fungsi primer yaitu memantau data pada robot, menentukan jaringan komunikasi yang digunakan, serta dapat menentukan misi yang sudah direncanakan dan sebagainya [11].

2.4 Graphical User Interface (GUI)

Graphical User Interface (GUI) merupakan tipe antarmuka yang digunakan oleh *user* agar dapat berinteraksi dengan sistem operasi melalui gambar-gambar grafik, ikon, menu dan menggunakan perangkat petunjuk seperti *mouse*, atau dll.

Manfaat dari mendesain GUI pada suatu sistem antarmuka yaitu dapat mengurangi komplikasi dan kesulitan pada pengguna dalam berkomunikasi dengan perangkat, dan mendesain GUI dapat memberikan nilai estetika lebih pada aplikasi yang dirancang [11].

Graphical User Interface merupakan salah satu media yang digunakan komputer sebagai cara untuk berkomunikasi dengan pengguna. Tujuan dari penggunaan *user interface* ini yaitu mengkomunikasikan fitur-fitur yang ada pada sistem agar pengguna bisa mengoperasikan sistem tersebut secara efisien dan tepat [12].

Ada juga beberapa kriteria yang harus diperhatikan dalam *graphical user interface* ini menurut, antar lain *significant design, development*, serta *testing activities*. Dengan menggunakan user interface tampilan *software* yang digunakan tentunya akan lebih bervariasi, menarik, interaktif, dan tentunya memberikan estetika lebih pada sebuah aplikasi [2].

2.5 Real Time System

Sistem *real-time* merupakan sistem yang kebenarannya secara logis didasarkan pada kebenaran hasil keluaran sistem dan ketepatan waktu hasilnya. *Real-time* juga merupakan sistem yang menghasilkan *output* atau respon yang tidak melebihi batas waktu yang telah ditetapkan. Menurut fungsinya, sistem ini terbagi menjadi *hard real-time system* dan *soft real-time system*. *Hard real-time system* diperlukan untuk menyelesaikan *critical task* dalam jaminan waktu tertentu. Biasanya, sebuah proses dikirim dengan sebuah pernyataan jumlah waktu dimana dibutuhkan untuk menyelesaikan atau menjalankan proses *input-output*. *Soft real-time system* memiliki sedikit kebebasan dibandingkan dengan sistem hard real-time karena software hanya mengutamakan pemrosesan, sehingga resiko kegagalan tidak fatal. Keterlambatan dalam waktu respons mengakibatkan penurunan kinerja dan penundaan yang lebih lama [13].

Suatu sistem dianggap *real-time* ketika tidak hanya memprioritaskan keakuratan atau ketepatan pelaksanaan instruksi ataupun tugas, tetapi juga interval

waktu dimana tugas dieksekusi. Dengan kata lain, sistem *real-time* adalah sistem yang menggunakan *deadline*, yaitu pekerjaan harus diselesaikan dalam waktu tertentu. Namun, sistem yang tidak *real-time* merupakan sistem yang tidak memiliki *deadline*, meskipun tentunya tetap mengharapkan respon yang cepat atau performa yang tinggi [14].

2.6 GPS

GPS merupakan sistem untuk menentukan letak di permukaan bumi dengan bantuan penyalarsan sinyal satelit, sistem navigasi yang menggunakan satelit yang didesain agar dapat menyediakan posisi secara instan, kecepatan dan informasi waktu di hampir semua tempat di muka bumi, setiap saat dan dalam kondisi cuaca. Tujuan awalnya dibuat GPS yaitu untuk keperluan militer, baik menunjukkan posisi secara tepat, dan membantu dalam memberikan arah [15].

GPS atau yang biasa disebut juga dengan *Global Positioning System* merupakan suatu sistem satelit navigasi dan penentuan posisi. Sistem ini didesain untuk memberikan posisi serta informasi waktu. GPS dapat digunakan untuk mengetahui posisi dan lokasi seseorang. Dengan adanya sensor GPS pada *payload*, maka GPS akan dimanfaatkan sebagai fitur *tracking* dalam memantau posisi *payload*. Yang mana nanti *payload* akan mengirimkan *longitude* dan *latitude* yang didapatkan dari sensor GPS yang bernama GPS Sensor Neo 6M dengan menggunakan radio telemetri. Kemudian telemetri *receiver* pada *ground control station* akan menerima *longitude* dan *latitude*. Yang kemudian *longitude* dan *latitude* tersebut akan diinput ke dalam google maps.net dan akan langsung menunjukkan dimana letak ataupun lokasi *payload* berada [16].

2.7 API

Application Programming Interface (API) menyediakan konsep fungsionalitas API, yang memungkinkan akses dan penggunaan aplikasi oleh pihak lain, serta komunikasi antar sistem walaupun berbeda *platform*. Proses tersebut dapat diakses dengan menggunakan arsitektur *representational state transfer* atau yang bisa disebut juga sebagai ReST. Yang dioperasikan menggunakan bahasa pemrograman *Hypertext Transfer Protocol* (HTTP) dan juga Java Script. API yang

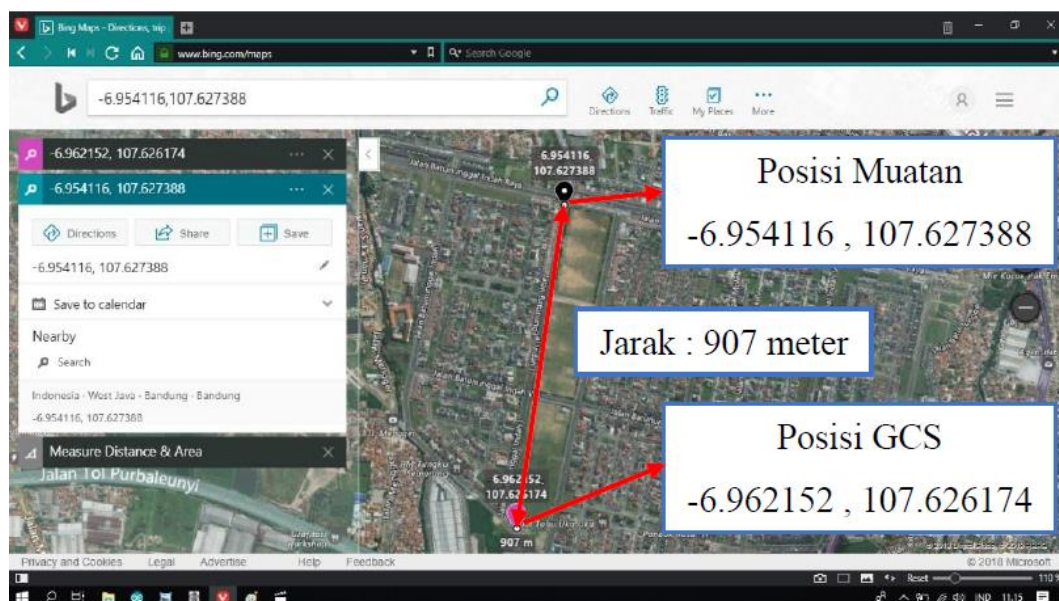
tersedia sebagai layanan web sering disebut juga sebagai API publik. Pengembang yang ini mencari publik API harus mencari melalui berbagai cara seperti mesin pencari umum, dokumentasi repositori, ataupun langsung di artikel web [17].

2.8 Google Map API

Google Maps merupakan layanan *mapping online* yang disediakan oleh google, untuk mengakses layanan ini dapat melalui situs <http://maps.google.com>. Pada google maps ini pengguna dapat melihat informasi geografis pada hampir seluruh wilayah di bumi. Layanan ini sangat nyaman digunakan karena di dalamnya pengguna dapat menggeser peta, memperbesar atau memperkecil peta, serta mengubah tampilan peta sesuai yang diinginkan oleh pengguna. Google maps juga membantu pengguna untuk menjelajah peta, mencari lokasi yang ingin dilihat atau dituju, seperti restaurant, tempat wisata, serta menampilkan rute perjalanan [18].

Google Maps API merupakan suatu *platform* yang disediakan dan dikembangkan oleh google, agar *developer* dapat mengintegrasikan peta, rute, dan tempat yang ada pada google maps ke dalam aplikasi yang dikembangkan oleh *developer* secara *real-time* [19]. Pada aplikasi monitoring profile vertikal atmosfer ini, API google maps berfungsi untuk *tracking payload* yang berada di udara.

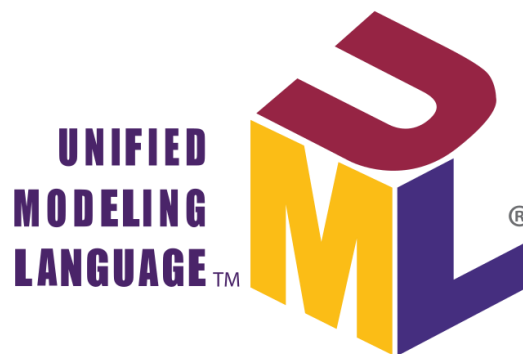
2.9 Jarak Komunikasi



Gambar 2. 1 Jarak Komunikasi

Pada Gambar 2.1 hasil pengujian dari penelitian sebelumnya mengenai jarak komunikasi *payload* dan *ground control station*. Data dapat dikirimkan dengan baik oleh *payload* dan diterima serta ditampilkan oleh *ground control station*, dengan jarak pengujian antara *payload* dan *ground control station* yaitu 907 meter [20].

2.10 Unified Modeling Language (UML)



Gambar 2. 2 Unified Modeling Language

UML merupakan salah satu *tool* / model yang berfungsi untuk merancang pengembangan *software* yang berorientasi objek. UML sendiri juga memberikan standar penulisan sebuah sistem yang disebut dengan *blueprint*, yang meliputi konsep proses bisnis, penulisan kelas-kelas dalam bahasa pemrograman yang spesifik, skema *database*, serta komponen apa saja yang diperlukan oleh *software* [21].

Seperti bahasa-bahasa lainnya, UML mendefinisikan notasi dan *syntax* / semantik. Notasi UML merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram piranti lunak, yang mana setiap bentuk mempunyai makna tertentu, dan UML *syntax* mendefinisikan bagaimana bentuk-bentuk tersebut dapat dikombinasikan atau dihubungkan. Notasi UML terutama diturunkan dari 3 notasi yang telah ada sebelumnya yaitu Gray Booch OOD (*Object Oriented Design*), Jim Rumbaugh OMT (*Object Modeling Technique*), dan Ivar Jacobson OOSE (*Object Oriented Software Engineering*). UML juga

mendefinisikan diagram-diagram seperti *use case diagram*, *class diagram*, *sequence diagram*, *collaboration diagram*, *activity diagram* [22].

2.11 Latitude dan Longitude

Garis Lintang atau *latitude* merupakan garis khayalan yang membelah bumi secara horizontal menjadi dua bagian yaitu utara dan selatan. Garis ini memiliki posisi membentangi bumi, sama halnya seperti garis khatulistiwa. Garis lintang 0 derajat membelah bumi menjadi dua bagian secara horizontal dan di atasnya disebut Lintang Utara, sedangkan di bawahnya disebut Lintang Selatan. Besaran tertinggi garis lintang adalah 90 derajat, diukur dalam derajat, menit, dan detik atau derajat desimal, utara atau selatan dari khatulistiwa. Garis lintang juga digunakan untuk mengukur koordinat suatu titik di belahan bumi, dengan nilai koordinat di bagian utara selalu positif dan nilai koordinat di bagian selatan selalu negatif. Dengan kata lain, garis lintang adalah garis yang menentukan lokasi berada di sebelah utara atau selatan ekuator.

Garis bujur atau *longitude* adalah garis khayalan yang membelah bumi secara vertikal menjadi dua bagian, yaitu bagian timur dan barat, dan menghubungkan Kutub Utara dan Selatan. Garis Bujur digunakan untuk mengukur sisi Barat-Timur koordinat suatu titik di belahan bumi dan dinyatakan dalam derajat, menit, dan detik. Garis tengah yang bernilai 0 derajat disebut garis *Prime Meridian*, sedangkan yang paling kiri memiliki nilai -90 derajat dan yang paling kanan memiliki nilai 90 derajat. *Longitude* juga dibedakan menjadi dua wilayah, yaitu Bujur Timur dan Bujur Barat, dimana koordinat yang berada di timur selalu bernilai negatif, dan sebaliknya yang berada di barat selalu positif. *Longitude* digunakan untuk menentukan lokasi di wilayah Barat atau Timur dan biasanya dilambangkan dengan huruf Yunani yaitu lambda (λ) [23][24][25].

2.12 Metode *Haversine*

Metode *Haversine* adalah sebuah sistem navigasi yang menggunakan rumus matematika untuk mencari jarak terpendek antara dua titik pada permukaan bola bumi, dengan menggunakan garis bujur dan lintang atau *longitude* dan *latitude* sebagai variabel *inputan*. Rumus *Haversine* ini pertama kali ditemukan oleh Jamez Andrew pada tahun 1805 dan kemudian dipergunakan oleh Josef de Mendoza y Rios pada tahun 1801. Prof. James Inman memperkenalkan istilah “*Haversine*” pada tahun 1835 dan istilah ini digunakan oleh Josef de Mendoza y Rios dalam penelitiannya mengenai masalah utama *Astronomi Nautical*. Selain digunakan untuk mengukur jarak antar bintang, rumus *Haversine* ini juga sangat akurat dalam menghitung jarak antara dua titik permukaan bumi, karena memperhitungkan bidang lengkung dan tidak memperhitungkan bukit dan lembah pada permukaan bumi. Dengan menggunakan asumsi bahwa bumi berbentuk bulat sempurna dengan jari-jari sebesar 6.367,45 km, rumus *Haversine* dapat memberikan jarak lingkaran besar antara dua titik pada permukaan bola, berdasarkan garis bujur dan lintang dari masing-masing titik tersebut. Pada penelitian ini rumus *Haversine* digunakan untuk menentukan ke arah mana objek menuju [26][27][28].

Rumus *Haversine* :

$$\Delta \text{ lat} = \text{lat}2 - \text{lat}1 \quad (1)$$

$$\Delta \text{ long} = \text{long}2 - \text{long}1 \quad (2)$$

$$a = \sin^2(\Delta \text{lat}/2) + \cos(\text{lat}1) \cdot \cos(\text{lat}2) \cdot \sin^2(\Delta \text{long}/2) \quad (3)$$

$$c = 2 \cdot \text{atan2}(\sqrt{a}, \sqrt{1-a}) \quad (4)$$

$$d = R \cdot c \quad (5)$$

Rumus Bearing:

$$y = \sin(\text{radLon}2 - \text{radLon}1) * \cos(\text{radLat}2) \quad (6)$$

$$x = \cos(\text{radLat}1) * \sin(\text{radLat}2) - \sin(\text{radLat}1) * \cos(\text{radLat}2) * \cos(\text{radLon}2 - \text{radLon}1) \quad (7)$$

$$\text{sudut} = \text{atan2}(y, x) * 180 / \Pi \quad (8)$$

Penjelasan:

R = jari-jari bumi sebesar 6371 (km)

$\Delta \text{ lat}$ = besaran perubahan latitude

$\Delta \text{ long}$ = besaran perubahan longitude

c = kalkulasi perpotongan sumbu

d = jarak (km)

1 = 0.0174532925 radian