



# PENGEMBANGAN APLIKASI GROUND CONTROL STATION (GCS) UNTUK PENGAWASAN DAN PENGENDALIAN UAV

Tazkia Karima Herli Efison<sup>1\*</sup>, Wahyu Eko Sulistiono<sup>2</sup>, Mona Arif Muda Batubara<sup>3</sup>, Gigih Forda Nama<sup>3</sup>

<sup>1,2,3</sup> Teknik Informatika Universitas Lampung; Jl. Prof. Dr. Ir. Sumantri Brojonegoro No. 1, Bandar Lampung, Lampung

Riwayat artikel:

Received: 29 September 2022

Accepted: 8 November 2022

Published: 15 Januari 2023

## Keywords:

Unmanned Aerial Vehicle (UAV), Ground Control Station (GCS), Mission Planner

## Correspondent Email:

tatakia07@gmail.com

## How to cite this article:

Tazkia (2023).  
Pengembangan Aplikasi  
Ground Control Station  
(GCS) Untuk Pengawasan  
dan Pengendalian UAV.  
*Jurnal Informatika dan  
Teknik Elektro Terapan*,  
11(1).

© 2023 JITET (Jurnal Informatika dan Teknik Elektro Terapan). This article is an open-access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY NC)

**Abstrak.** Aplikasi *Mission Planner* merupakan salah satu aplikasi *Ground Control Station* yang bersifat *open-source*. Aplikasi *Mission Planner* dibangun oleh Michael Osborne dan menjadi aplikasi GCS yang paling umum digunakan. Aplikasi *Mission Planner* digunakan untuk mengontrol serta memonitoring misi penerbangan. Aplikasi ini juga dapat menerima serta mengirim perintah data penerbangan. Dalam melakukan misi penerbangan, aplikasi *Mission Planner* dioperasikan oleh operator GCS. Dalam mengoperasikan *Mission Planner*, akan lebih baik jika tampilan aplikasi disesuaikan dengan kebutuhan operator. Oleh karena itu, dilakukan pengembangan tampilan aplikasi *Mission Planner* yang sesuai dengan kebutuhan. Sehingga diharapkan dapat mempermudah operator GCS dalam mengoperasikan penerbangan. Penelitian ini mengembangkan tampilan dari aplikasi *Mission Planner* berbasis desktop. Pengembangan dilakukan dengan menggunakan *Microsoft Visual Studio Community 2019* dengan bahasa pemrograman C#. Penelitian ini diuji coba dengan menggunakan fasilitas dari Laboratorium Teknik Digital.

**Abstract.** *Mission Planner* is an open-source software of Ground Control Station. The *Mission Planner* was built by Michael Osborne and is the most commonly used GCS software. This software is used to control and monitor flight missions. This software also able to receive and send flight data commands. In carrying out flight missions, the *Mission Planner* application is operated by the GCS operator. In operating *Mission Planner*, it would be better if the appearance of this software is adjusted to the needs of the operator. Therefore, the *Mission Planner* application display was developed according to the needs. Hopefully, this development will make it easier for GCS operators to operate flights. This study develops the appearance of the desktop-based *Mission Planner* software. The development is carried out using *Microsoft Visual Studio Community 2019* with the C# programming language. This research was tested using facilities from the Digital Engineering Laboratory.

## 1. PENDAHULUAN

*Ground Control Station* (GCS) merupakan pusat kontrol di darat yang terdiri dari beberapa fasilitas untuk mengontrol kendaraan tanpa awak di udara. GCS dapat

digunakan untuk menerima data maupun mengirim perintah. Ada beberapa aplikasi *Ground Control Station* yang biasa digunakan untuk melakukan penerbangan. Namun, salah satu aplikasi GCS yang paling umum digunakan

adalah aplikasi *Mission Planner*. Aplikasi *Mission Planner* merupakan aplikasi *Ground Control Station* yang bersifat *Open Source* dan dapat disesuaikan dengan kebutuhan pengguna, baik dari segi fitur, sistem, maupun dari segi tampilannya.

Berawal dari adanya aplikasi tersebut dan bersifat open source, muncul lah ide untuk mengembangkan aplikasi *Mission Planner* dengan perubahan tampilan yang sesuai dengan kebutuhan. Pengembangan tampilan yang dilakukan akan melalui pendekatan *User Experience*, dimana pengalaman user akan berperan penting dalam pengembangan aplikasi ini.

Diharapkan dengan adanya pengembangan aplikasi ini dapat mempermudah pengguna dalam melakukan misi penerbangan. Dengan begitu, penelitian ini dilakukan untuk mengembangkan tampilan aplikasi *Mission Planner* berbasis *desktop*. Aplikasi ini dikembangkan dengan menggunakan aplikasi *Microsoft Visual Studio Community 2019* dengan bahasa pemrograman *C#.NET*. Penelitian ini akan diuji coba dengan menggunakan fasilitas milik Laboratorium Teknik Digital.

## 2. TINJAUAN PUSTAKA

### 2.1 Unmanned Aerial Vehicle (UAV)

Unmanned Aerial Vehicle (UAV) adalah pesawat yang dikendalikan tanpa awak dan dapat berfungsi dengan adanya kendali jarak jauh oleh pilot. UAV menggunakan hukum aerodinamika untuk melakukan penerbangan. UAV mampu membawa muatan seperti senjata ataupun muatan lainnya. UAV memiliki ukuran, bentuk, karakter, serta konfigurasi yang bervariasi. UAV dapat dikontrol melalui alat pengendali jarak jauh serta dapat terbang berdasarkan perintah misi yang telah dimasukkan ke dalam sistem [1].

### 2.2 Ground Control Station (GCS)

*Ground Control Station* (GCS) merupakan stasiun kontrol darat yang berfungsi sebagai pengendali dan pemantau UAV yang sedang melaksanakan misi penerbangan secara jarak jauh dari darat. *Ground Control Station* terdiri dari tiga bagian utama, yaitu *Mission*

*Control System* (MCS), *Payload Control System* (PCS), dan *Communication System* (CS).

*Mission Control System* (MCS) merupakan aplikasi *Ground Control Station* (GCS) yang terinstall pada komputer baik *Personal Computer* (PC), laptop, *Raspberry Pi*, maupun *mini computer*. Aplikasi ini dioperasikan oleh *Ground Control Operator* (GCO) atau dalam bahasa Indonesia biasa disebut *Operator GCS*.

*Mission Control System* menampilkan semua informasi yang diperlukan pada *Operator GCS* sehingga operator tidak perlu bergantung pada bantuan eksternal. Data yang bisa dipantau oleh *operator GCS* contohnya seperti *indication airspeed*, *ground speed*, posisi pesawat, *horizon virtual*, status bahan bakar, status persentase baterai, arah, ketinggian, kecepatan dan arah angin, dan sebagainya.

Pada penelitian ini, *Mission Control System* (MCS) diinstall pada laptop sebagai aplikasi utama GCS. *Communication System* (CS) berbentuk *antenna* yang terhubung pada laptop sebagai *transmitter* dan pada UAV sebagai *receiver*. Modul antena yang digunakan adalah *Module Telemetry 433 MHz*.

### 2.3 Mission Planner

*Mission Planner* merupakan salah satu aplikasi *Ground Control Station* yang digunakan untuk melakukan mengendalikan dan memonitoring UAV selama misi penerbangan berlangsung. *Flight Controller* berfungsi untuk mengatur kendali serta stabilisasi UAV.

Sedangkan *Mission Planner* digunakan untuk mengatur *waypoint* yang nantinya harus dilewati oleh UAV serta memantau data-data yang didapatkan dari hasil uji terbang menggunakan *Flight Controller*. *Monitoring* ini perlu dilakukan guna untuk mengawasi keadaan UAV selama melakukan penerbangan agar keadaan tetap stabil.

Dengan begitu misi yang dijalankan juga dapat berjalan dengan baik. Aplikasi *Mission Planner* ini merupakan aplikasi desktop yang bersifat *Open Source*. Aplikasi ini dikembangkan oleh Michael Osborne.

### 2.4 Penelitian Terkait

Penelitian terkait pertama yaitu oleh Aris Susilo dengan judul Pengembangan Sistem

*Autopilot* Pada Wahana Udara Tanpa Awak *Fixed Wing* Proto-03. Penelitian ini mengembangkan sistem *autopilot* dengan melakukan konfigurasi aplikasi dengan perangkat keras berupa sebuah pesawat *aeromodelling*. Sehingga pesawat tersebut dapat melakukan penerbangan secara otomatis dengan mengikuti *waypoint* yang telah ditentukan [2].

Penelitian terkait kedua yaitu oleh Ariesta dan Isnan dengan judul *Sistem Ground Control Station* berbasis *Mobile* untuk Pengamatan dan Pengendalian UAV. Penelitian ini dirancang dapat menampilkan parameter penerbangan UAV melalui *Android*. Data tersebut telah disesuaikan dengan data yang didapat dari modul ADAHRS. Dalam implementasinya, *GCS mobile* dan *quadcopter* akan saling terhubung dengan menggunakan *transmitter* dan *receiver*. Kemudian data status *quadcopter* akan ditampilkan pada layar *Android*. Aplikasi ini dibangun untuk memantau dan mengendalikan UAV dari jarak jauh ketika sudah berada di luar jangkauan pandangan [3].

Penelitian terkait ketiga yaitu mengenai *Ground Control Station Development for Autonomous UAV* oleh Ye Hong, Jiancheng Fang, dan Ye Tao ini dikembangkan menggunakan bahasa pemrograman C++. Pada GUI utama menampilkan *toolbar*, *flight path*, *equipment monitoring*, *instrument board*, *information of UAV*, dan *serially transmitted data*. Pada penelitian ini dibangun aplikasi GCS yang terdiri dari tiga hal utama, yakni GCS yang berfungsi dengan baik, perangkat keras, dan panel instrumen untuk menampilkan informasi, jalur penerbangan, dan lain-lain [4].

Penelitian terkait keempat oleh Ariesta Martiningtyas Handayani dan Budi Sumanto membangun Sistem *Ground Control Station* untuk Pengamatan dan Pengendalian *Unmanned Aerial Vehicle*. Sistem ini dibangun untuk mendeteksi GPS, sensor IMU (*Inertial Measurement Unit*) yang terdiri *accelerometer*, *gyroscopes*, dan *magnetometer* yang diletakkan pada UAV. GUI dirancang menggunakan Visual Studio C#, GIS didesain dengan Gmap.NET, pemrosesan video dengan EmguCV, dan *database* menggunakan MySQL. Sistem ini dibuat untuk menampilkan lokasi UAV pada *desktop* serta data keadaan UAV seperti *altitude*, posisi (*Latitude* dan *longitude*),

arah, *roll*, *pitch*, *yaw*. Selain itu, terdapat fitur *video streaming* yang dapat digunakan ketika UAV telah dipasangkan kamera [5].

### 3. METODE PENELITIAN

*Prototype* adalah salah satu pendekatan dalam rekayasa aplikasi yang secara langsung mendemonstrasikan bagaimana sebuah aplikasi akan bekerja dalam lingkungannya sebelum tahapan implementasi dilakukan. Pengembang dan pengguna bertemu dan mendefinisikan obyektif keseluruhan dari aplikasi dan mengidentifikasi segala kebutuhan yang diketahui. Kunci dari *prototype* adalah mendefinisikan aturan dan kebutuhan di awal, yakni pengguna dan pengembang sama-sama penggunaan *prototype* digunakan sebagai mekanisme pendefinisian kebutuhan [6].

Dalam model *prototype*, perangkat lunak yang dihasilkan nantinya akan dipresentasikan kepada pengguna, dan pengguna akan diberikan kesempatan untuk memberikan masukan sehingga perangkat lunak yang dihasilkan nantinya akan sangat sesuai dengan keinginan dan kebutuhan pengguna.

Tahapan model *prototype* dapat dijabarkan sebagai berikut :

#### a. Requirement and Analysis

Pada tahap ini, pengembang dan customer bertemu dan mendefinisikan keseluruhan perangkat lunak yang diinginkan masyarakat, mengidentifikasi kebutuhan, mengidentifikasi masalah dari customer serta mendefinisikan tujuan ke depan dari pengembangan *software* dan sasaran yang akan dicapai.

#### b. Quick Design and Modelling Quick Design

Tahapan ini melanjutkan proses dari tahap *communication* yaitu membuat perencanaan dan permodelan secara cepat lebih difokuskan dalam mempresentasikan aspek-aspek apa saja yang akan ditampilkan atau dilihat oleh masyarakat.

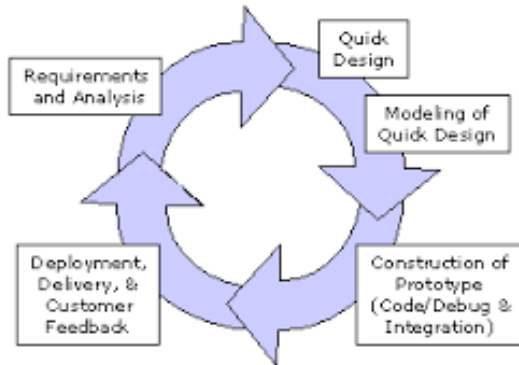
#### c. Construction of Prototype

Tahapan ini merupakan proses pembuatan program yang berdasarkan spesifikasi sistem dan desain perancangan yang didapatkan dari hasil tahapan sebelumnya.

#### d. Deployment Delivery and Feedback

Hasil dari pembuatan program akan diberikan ke masyarakat dan kemudian dievaluasi oleh masyarakat sendiri. *Feedback*

yang didapatkan dari keluhan ataupun saran yang diberikan oleh masyarakat setelah mengevaluasi *prototype* yang telah dicoba.



Gambar 1. Metode Prototype

## 4. HASIL DAN PEMBAHASAN

### 4.1 Requirement and Analysis

Pada tahap ini dilakukan pencarian referensi yang merujuk pada penelitian. Pengkajian dilakukan pada hasil pencarian referensi sebagai bahan dasar acuan, antara penelitian yang telah ada dengan penelitian saat ini.

Pada tahap ini didapatkan referensi dokumentasi untuk mengembangkan aplikasi *Mission Planner* dengan judul “*Building Mission Planner with Visual Studio*” pada website <https://ardupilot.org/dev/>. *Mission Planner* terdiri dari beberapa *project* sebagai berikut.

Tabel 1. Project Mission Planner

No	<i>Project Mission Planner</i>
1	<i>MissionPlanner (main code)</i>
2	<i>AviFile</i>
3	<i>BaseClasses</i>
4	<i>BSE.Windows.Forms</i>
5	<i>Core</i>
6	<i>GeoUtility</i>
7	<i>GMap.Net.Core</i>
8	<i>GMap.Net.WindowsForms</i>
9	<i>KMLib</i>
10	<i>MAVLink</i>
11	<i>MetadataExtractor</i>
12	<i>MissionPlanner.Comms</i>
13	<i>MissionPlanner.Controls</i>

14	<i>MissionPlanner.Utils</i>
15	<i>px4uploader</i>
16	<i>SharpKml</i>
17	<i>ZedGraph</i>

#### 4.1.1 Analisa Kebutuhan

Pada tahap analisa kebutuhan, pengguna dan pengembang bersama-sama mendefinisikan format seluruh aplikasi, mengidentifikasi semua kebutuhan, dan garis besar sistem yang akan dibuat. Tahap ini terdiri dari Definisi Masalah, Solusi Alternatif, Penentuan Target Diskusi, Verifikasi Penggunaan Metode, dan Pengumpulan Data Kebutuhan Pengguna.

##### a. Definisi Masalah

*Mission Planner* merupakan salah satu aplikasi *Ground Control Station* (GCS) berbasis desktop yang umum digunakan. Penelitian ini akan mengembangkan tampilan aplikasi sesuai dengan yang dibutuhkan guna mempermudah dalam melakukan misi penerbangan.

##### b. Solusi Alternatif

Mengembangkan Aplikasi *Ground Control Station* yang diharapkan dapat membantu tim dalam melaksanakan misi dengan lebih baik.

##### c. Penentuan Target Diskusi

Target Diskusi untuk Pengumpulan Data Kebutuhan Pengguna adalah dengan salah satu Operator GCS.

##### d. Verifikasi Penggunaan Metode

Metode yang digunakan dalam mengembangkan Aplikasi GCS adalah Metode *Prototype*. *Prototype* berfungsi untuk membuat perancangan sementara yang berfokus pada penyajian kepada pengguna.

##### e. Pengumpulan Data Kebutuhan

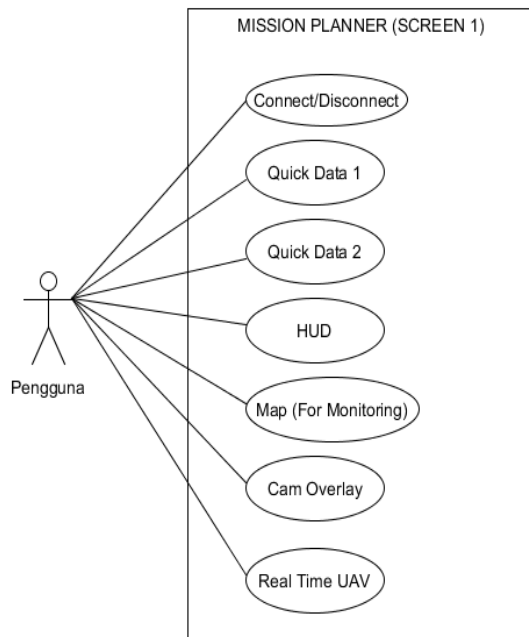
Pada tahap ini dilakukan Pengumpulan Data Kebutuhan Pengguna.

Tabel 2. Kebutuhan Tampilan pada Aplikasi

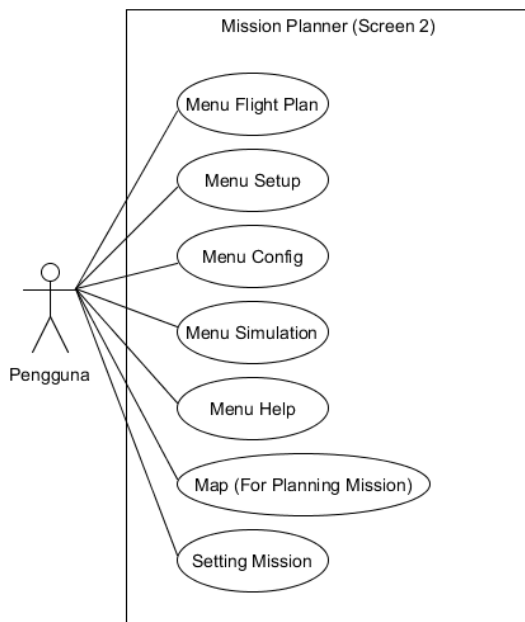
No.	Fitur	Keterangan
1.	Screen 1 (Flight Data)	HUD
		Map Area
		Quick Data 1 (Status)
		Quick Data 2 (Status)
		Port Connection
		Cam Overlay
		Realtime Status of UAV (With UAV Model)

2.	Screen 2 (Flight Plan)	Maps
		Setting Mission
		Menu Initial Setup
		Menu Config/Tuning

#### 4.1.2 Use Case Diagram



Gambar 2. Usecase Flight Data



Gambar 3. Usecase Flight Planner

#### 4.1.3 Use Case Scenario

Terdapat beberapa skenario yang akan dideskripsikan pada pembahasan ini.

Tabel 3. Melakukan Connect UAV

<b>MISSION PLANNER</b>	<b>Connect UAV</b>
Ringkasan	Pengguna melakukan <i>connect</i> antara aplikasi dengan UAV
Rasional	<i>Usecase</i> agar pengguna dapat melakukan <i>Connect</i>
Pengguna	<i>Operator GCS</i>
Prekondisi	-
Aliran Dasar	<ol style="list-style-type: none"> <li>1. Pengguna memasang <i>telemetry (transmitter)</i> pada USB laptop</li> <li>2. Pengguna memasang <i>telemetry (receiver)</i> pada <i>Flight Controller</i> yang terpasang di UAV</li> <li>3. Pengguna menghidupkan UAV dan membuka aplikasi <i>Mission Planner</i></li> <li>4. Pengguna mengatur <i>Port COM</i> serta <i>Baud Rate</i> yang sesuai dengan <i>telemetry</i></li> <li>5. Pengguna menekan tombol "<i>Connect</i>"</li> <li>6. Memproses koneksi UAV dengan aplikasi</li> <li>7. UAV telah terkoneksi dengan aplikasi ditandai dengan adanya status keadaan UAV serta lokasi UAV pada map</li> </ol>
Aliran Alternatif	-
Post Kondisi	Dapat melakukan <i>Connect</i>

Tabel 4. Melihat Status Data UAV

<b>MISSION PLANNER</b>	<b>Melihat Status Data UAV</b>
Ringkasan	Pengguna melihat status data UAV
Rasional	<i>Usecase</i> agar melihat status data UAV
Pengguna	<i>Operator GCS</i>
Prekondisi	Pengguna sudah melakukan koneksi UAV dengan aplikasi
Aliran Dasar	<ol style="list-style-type: none"> <li>1. Pengguna melihat status data UAV pada <i>Screen 1</i> (Sebelum misi dilakukan)</li> </ol>

	2. Memproses data dari <i>Flight Controller</i>
	3. Pengguna dapat melihat informasi status data UAV <ul style="list-style-type: none"> <li>· <i>Quick Data 1</i></li> <li>· <i>Quick Data 2</i></li> <li>· HUD</li> <li>· Map</li> <li>· <i>Real Time</i> UAV</li> </ul>
Aliran Alternatif	-
Post Kondisi	Dapat melihat informasi status data UAV

Tabel 5. Mengakses Menu Pada Aplikasi

<b>MISSION PLANNER</b>	<b>Mengakses Seluruh Menu pada Aplikasi</b>
Ringkasan	Pengguna mendapatkan informasi dari setiap menu
Rasional	<i>Usecase</i> agar pengguna mendapatkan informasi
Pengguna	<i>Operator GCS</i>
Prekondisi	Pengguna sudah melakukan koneksi UAV dengan aplikasi
Aliran Dasar	1. Pengguna menekan <i>icon</i> menu 2. Memproses data dari <i>database</i> 3. Informasi ditampilkan di layar
Aliran Alternatif	-
Post Kondisi	Dapat mengakses seluruh menu diaplikasi

Tabel 6. Membuat Misi Penerbangan

<b>MISSION PLANNER</b>	<b>Membuat Misi Penerbangan</b>
Ringkasan	Pengguna membuat misi penerbangan pada <i>Screen 2</i> sesuai dengan misi yang dijalankan
Rasional	<i>Usecase</i> untuk membuat misi penerbangan pada <i>Screen 2</i>
Pengguna	<i>Operator GCS</i>
Prekondisi	Pengguna sudah melakukan koneksi UAV dengan aplikasi

Aliran Dasar	1. Pengguna membuat misi penerbangan pada <i>Screen 2</i> dengan membuat jalur penerbangan ( <i>waypoint</i> ), aksi yang akan dilakukan, dan sebagainya 2. Misi penerbangan akan tersimpan pada <i>Flight Controller</i> yang terpasang di badan UAV 3. Pengguna dapat mengedit misi penerbangan sesuai dengan kebutuhan 4. Data terbaru dari misi penerbangan akan ditampilkan pada <i>Flight Data</i> di <i>Screen 1</i>
Aliran Alternatif	-
Post Kondisi	Dapat melakukan <i>setting</i> misi penerbangan sesuai kebutuhan

Tabel 7. Melakukan Setting Awal UAV

<b>MISSION PLANNER</b>	<b>Melakukan Setting Awal UAV</b>
Ringkasan	Pengguna melakukan <i>setting</i> awal UAV, mulai dari <i>Firmware</i> , kalibrasi, serta <i>setting</i> perangkat yang akan dipakai seperti <i>telemetry</i> atau <i>antenna tracker</i> .
Rasional	<i>Usecase</i> agar melakukan <i>setting</i> awal UAV
Pengguna	<i>Operator GCS</i>
Prekondisi	Pengguna sudah mengkoneksikan UAV dengan aplikasi
Aliran Dasar	1. Pengguna memilih menu <i>Setup</i> pada <i>Screen 2</i> 2. Pengguna melakukan konfigurasi awal pada <i>Flight Controller</i> serta komponen lainnya yang digunakan 3. <i>Setting</i> awal akan tersimpan di <i>Flight Controller</i>
Aliran Alternatif	-

Post Kondisi	Dapat melakukan <i>setting</i> awal UAV
--------------	-----------------------------------------

Tabel 8. Mengatur Parameter Sebelum Terbang

<b>MISSION PLANNER</b>	<b>Mengatur Parameter Sebelum Terbang</b>
Ringkasan	Pengguna mengatur parameter secara lengkap
Rasional	<i>Usecase</i> agar dapat mengatur parameter secara lengkap
Pengguna	<i>Operator GCS</i>
Prekondisi	-
Aliran Dasar	<ol style="list-style-type: none"> <li>1. Pengguna menekan <i>Menu Config</i> pada <i>Screen 2</i></li> <li>2. Pengguna mengatur parameter secara lengkap</li> <li>3. Pengguna <i>mengklik</i> simpan</li> <li>4. Data berhasil disimpan di <i>Flight Controller</i></li> </ol>
Aliran Alternatif	-
Post Kondisi	Dapat mengatur parameter secara lengkap

Tabel 9. *Disconnect UAV*

<b>MISSION PLANNER</b>	<b><i>Disconnect UAV</i></b>
Ringkasan	Pengguna memutus koneksi antara aplikasi dengan UAV
Rasional	<i>Usecase</i> agar pengguna dapat memutus koneksi antara aplikasi dengan UAV
Pengguna	<i>Operator GCS</i>
Prekondisi	Pengguna sudah melakukan koneksi antara aplikasi dengan UAV
Aliran Dasar	<ol style="list-style-type: none"> <li>1. Pengguna <i>mengklik</i> tombol <i>Disconnect</i></li> <li>2. Pengguna keluar dari aplikasi dan kembali ke halaman utama</li> </ol>
Aliran Alternatif	-
Post Kondisi	Dapat memutus koneksi antara aplikasi dengan UAV

## 4.2 Quick Design and Modelling of Quick Design

Pada tahap ini mulai merancang desain tampilan yang ingin dikembangkan dari aplikasi *Mission Planner* yang sesuai dengan kebutuhan pengguna. Sistem pada aplikasi ini akan berjalan pada Windows dan dikembangkan menggunakan bahasa pemrograman C# dengan menggunakan *tools Visual Studio Community 2019*. Berikut beberapa perancangan pada aplikasi ini:

### 4.2.1 Icon / Logo Aplikasi



Gambar 4. Logo Aplikasi

Pada logo aplikasi *Mission Planner* ini dilakukan penambahan logo URO disamping lambang *ArduPilot* sebagai identitas dari aplikasi.

### 4.2.2 Halaman *Splash Screen*

Gambar 5. *Splash Screen*

Pada halaman *Splash Screen* diberikan penambahan logo URO sebagai identitas dari aplikasi.

### 4.2.3 Halaman *Flight Data*

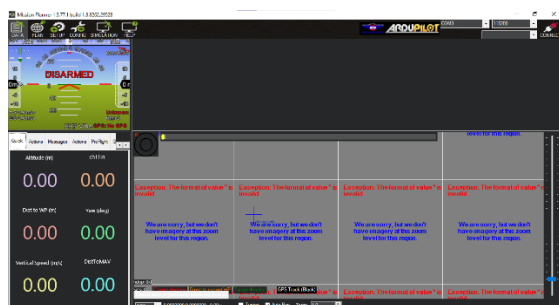
Pada halaman *Flight Data* ini dilakukan pengembangan dengan menampilkan kolom *Status*. Sehingga ketika misi sedang berjalan, *operator* tidak perlu berganti kolom dari *Quick Data* menjadi Kolom *Status*. Selain itu, kolom *Status* ini memiliki banyak sekali data sehingga dilakukan pengurangan beberapa *list* yang tidak diperlukan.



Gambar 6. Halaman *Flight Data*

Halaman *Flight Data* digunakan untuk memantau kondisi UAV pada saat sedang melakukan penerbangan (*On Mission*). Pada halaman *Flight Data* ini terdiri dari *Quick Data* 1, *HUD*, *Status*, *Connection Port*, *Map (On Mission)*, *Cam Overlay*, dan gambaran *Real Time UAV*.

Hasil Implementasi:

Gambar 7. Hasil Implementasi *Flight Data*

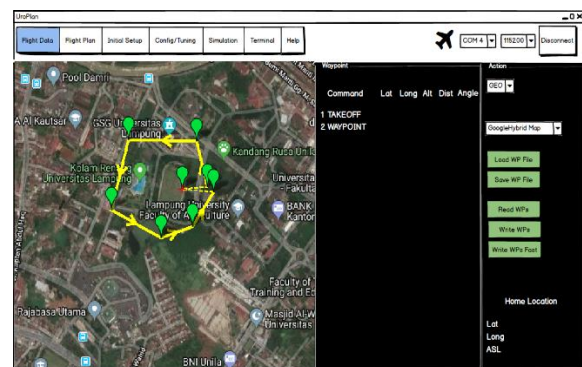
Pada Gambar 7. Hasil implementasi *Flight Data* pada halaman *Flight Data* telah berhasil diimplementasikan. Namun tidak semua fitur berhasil diimplementasikan. Hal ini disebabkan karena adanya beberapa keterbatasan, seperti waktu dan pengembang. Fitur yang telah berhasil diimplementasikan adalah menampilkan kolom *Status Data*. Untuk *RealTime UAV* serta *Camera Overlay* belum berhasil diimplementasikan. Berikut ini merupakan tabel penjabaran dari fitur yang telah berhasil dan belum berhasil dibuat.

Tabel 10. Fitur pada Halaman *Flight Data*

1	<i>Quick Data</i>	Berhasil
2	<i>Status Data</i>	Berhasil
3	<i>HUD</i>	Berhasil
4	<i>Maps</i>	Berhasil
5	<i>Camera Overlay</i>	Belum Berhasil
6	<i>Real Time UAV</i>	Belum Berhasil

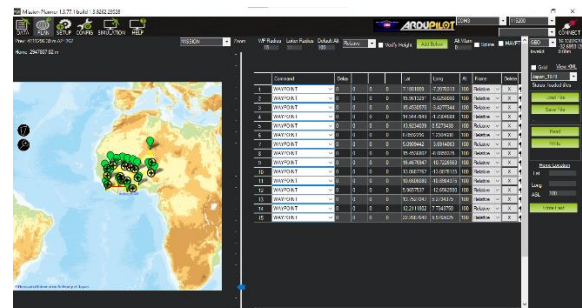
#### 4.2.4 Halaman *Flight Planner*

Pada halaman *Flight Planner* ini dilakukan pengubahan tampilan dimana kolom *list* dari titik *Waypoint* yang telah ditentukan akan ditampilkan pada kolom samping kanan. Hal bertujuan agar operator tidak perlu lagi melakukan *resize* kolom *waypoint* untuk melihat semua *list*. Selain itu, operator juga tidak perlu melakukan *scroll down/up* lagi.

Gambar 8. Halaman *Flight Planner*

Pada halaman *Flight Plan* ini terdapat beberapa fitur yang tertera pada halaman, yakni *Menu Flight Plan*, *Menu Setup*, *Menu Config*, *Map*, dan *Setting Waypoint*.

Hasil Implementasi:

Gambar 9. Implementasi *Flight Planner*



Berdasarkan Gambar 9. Hasil Implementasi Halaman *Flight Planner* bahwa seluruh fitur berhasil diimplementasikan pada *Flight Planner*. Berikut ini penjabaran dari fitur yang telah berhasil dibuat.

Tabel 11. Fitur pada Halaman *Flight Planner*

No.	Fitur	Berhasil / Belum Berhasil
1	<i>Maps</i>	Berhasil
2	<i>Setting Waypoint (Vertical)</i>	Berhasil
3	<i>Side setting lainnya</i>	Berhasil

### 4.3 Deployment, Delivery, & Customer Feedback

Tahap terakhir dari metode *prototyping* adalah pengujian aplikasi atau *testing*. Penelitian ini menggunakan metode *Blackbox Testing* untuk menguji aplikasi berdasarkan detail aplikasi seperti tampilan aplikasi, fungsi – fungsi pada aplikasi, serta kesesuaian alur fungsi serta tampilan dengan bisnis proses yang diinginkan pengguna.

Kemudian, penelitian ini juga menggunakan UEQ (*User Experience Quistionnaire*). Pengujian ini lebih menekankan pada poin *Physical Action*. *Physical Action* adalah pengujian yang dilakukan secara langsung kepada pengguna dengan menggunakan aplikasi. Seperti mencoba seluruh fitur pada aplikasi serta mengecek tampilan GUI (*Graphic User Interface*)

#### 4.3.1 Pengujian Blackbox

Sebelum dilakukan pengujian secara langsung kepada pengguna, dilakukan pengujian *Blackbox* terlebih dahulu. Pengujian *Blackbox* ini berfungsi untuk menguji apakah fungsi tetap berjalan dengan baik atau tidak. Capaian dari pengujian *Blackbox* adalah “Berhasil” atau “Memenuhi Syarat” artinya fungsi yang diuji sudah sesuai dengan kebutuhan. Berikut ini adalah daftar tabel dari pengujian *Blackbox* :

##### a. Pengujian Akses Aplikasi

Tabel 12. *Blackbox Testing* Akses Aplikasi

Input	Tujuan	Hasil	Kondisi
Menekan ikon atau logo	Pengguna dapat masuk ke halaman	Pengguna dapat masuk ke halaman	Berhasil

aplikasi pada PC	utama aplikasi	utama aplikasi	
------------------	----------------	----------------	--

##### b. Pengujian Connect Aplikasi

Tabel 13. *Blackbox Testing* Connect Aplikasi

Input	Tujuan	Hasil	Kondisi
Menekan tombol <i>Connect</i> pada aplikasi	Pengguna dapat melakukan koneksi antara aplikasi dengan <i>Flight Controller</i>	Pengguna berhasil melakukan koneksi aplikasi dengan <i>Flight Controller</i> dan status UAV tertera pada halaman	Berhasil

##### c. Pengujian Halaman *Flight Data*

Tabel 14. *Blackbox Testing* Akses Halaman *Flight Data*

Input	Tujuan	Hasil	Kondisi
Pengguna menekan tombol <i>Connect</i> pada halaman <i>Flight Data</i>	Pengguna dapat melihat informasi pada menu <i>Flight Data</i>	Pengguna dapat melihat informasi pada menu <i>Flight Data</i>	Berhasil

##### d. Pengujian Mengakses Menu Di App

Tabel 15. *Blackbox Testing* Akses Tiap Menu

Input	Tujuan	Hasil	Kondisi
Pengguna menekan ikon menu yang ingin di akses	Pengguna dapat melihat informasi pada menu yang dipilih	Pengguna dapat melihat informasi pada menu yang dipilih	Berhasil

##### e. Pengujian Halaman *Flight Plan*

Tabel 16. *Blackbox Testing* Akses Halaman *Flight Plan*

Input	Tujuan	Hasil	Kondisi
Pengguna menekan tombol <i>Connect</i> pada halaman	Pengguna dapat melihat informasi pada menu <i>Flight Plan</i> dan	Pengguna dapat melihat informasi pada menu <i>Flight Plan</i> dan	Berhasil

<i>Flight Data</i>	melakukan <i>setting waypoint</i>	melakukan <i>setting waypoint</i>	
--------------------	-----------------------------------	-----------------------------------	--

#### f. Pengujian Disconnect Aplikasi

Tabel 17. *Blackbox Testing Disconnect App*

Input	Tujuan	Hasil	Kondisi
Menekan ikon atau tombol “Disconnect”	Pengguna dapat memutuskan koneksi antara <i>Mission Planner</i> dengan <i>Flight Controller</i>	Pengguna dapat memutuskan koneksi antara <i>Mission Planner</i> dan <i>Flight Controller</i>	Berhasil

#### 4.3.2 Pengujian User Experience Questionnaire (UEQ)

Pengujian dilakukan dengan menggunakan metode UEQ untuk mengetahui pengalaman pengguna terhadap aplikasi yang di gunakan. Metode UEQ digunakan karena tampilan aplikasi di kembangkan dengan memperhatikan *User Experience*. Dengan begitu, metode yang tepat digunakan untuk mendapatkan *feedback User Experience* adalah dengan menggunakan UEQ. Metode UEQ juga sudah lengkap dan dapat diakses secara langsung pada *website www.ueq-online.org* dan tersedia dalam 36 bahasa. Berikut ini adalah tampilan kuesioner dalam Bahasa Indonesia :

Nama : .....  
 Jurusan : .....  
 Silakan evaluasi produk dengan memilih satu lingkaran tiap baris item.

	1	2	3	4	5	6	7	
menyusahkan	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	menyenangkan
tak dapat dipahami	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	dapat dipahami
kreatif	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	monoton
mudah dipelajari	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	sulit dipelajari
bermanfaat	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	kurang bermanfaat
membosankan	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	mengasyikkan
tidak menarik	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	menarik
tak dapat diprediksi	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	dapat diprediksi
cepat	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	lambat
berdaya cipta	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	konvensional
menghalangi	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	mendukung
baik	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	buruk
rumit	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	sederhana
tidak disukai	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	menggembirakan
lazim	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	terdepan
tidak nyaman	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	nyaman
aman	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	tidak aman
memotivasi	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	tidak memotivasi
memenuhi ekspektasi	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	tidak memenuhi ekspektasi
tidak efisien	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	efisien
tidak jelas	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	membingungkan
tidak praktis	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	praktis
terorganisasi	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	berantakan
atraktif	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	tidak atraktif
ramah pengguna	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	tidak ramah pengguna
konservatif	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	inovatif

CATATAN KRITIK DAN SARAN: .....

Gambar 10.1 UEQ Yang Digunakan

#### 4.3.3 Responden Kuisisioner

Penentuan responden berdasarkan pengguna yang sudah ahli dan sering menggunakan aplikasi ini dan merupakan *Operator GCS* pada organisasi URO.

#### 4.3.4 Hasil Dari Metode Prototyping

Total seluruh responden adalah 5 orang, akan tetapi 1 di antaranya termasuk dalam responden dinamis yang dalam proses pengembangan aplikasi ini terus menerus memberi masukan, sehingga saran dari responden diimplementasikan pada aplikasi ini. Hal ini bertujuan agar tampilan aplikasi yang dikembangkan lebih sesuai dengan keinginan pengguna. *Metode prototype* dilakukan dengan cara menguji aplikasi yang siap digunakan ke pengguna, kemudian merevisi berdasarkan saran dari responden yang dapat diimplementasikan. Hal ini dilakukan berulang kali hingga saran dari responden semakin terpenuhi. Dalam hal ini, berikut adalah list saran tampilan dari responden serta fitur - fiturnya:

Tabel 18. List Saran Responden

Meet	Saran	Perbaiki
1	<p>- Hapus <i>Menu Simulation</i> dan <i>Menu Help</i></p> <p>- Pisah tampilan aplikasi menjadi tiga bagian untuk tiga monitor (<i>Full Map</i>, <i>Full Status Data</i>, dan <i>Flight Plan</i>)</p> <p>- Halaman <i>Full Status Data UAV</i> terdiri dari <i>Quick Data</i>, <i>Status</i>, dan <i>Action</i></p> <p>- Halaman <i>Full Map</i> hanya berisikan <i>Map</i> saat UAV sedang <i>On Mission</i></p> <p>- Halaman <i>Flight Plan</i> berisikan <i>Map</i> untuk <i>setting misi</i>,</p>	<p>- Perbaiki <i>icon Ardupilot</i></p> <p>- Perbaiki Halaman <i>Full Status Data</i> dari <i>Quick Data</i>, <i>Status</i>, dan <i>Action</i></p> <p>- Perbaiki halaman <i>Full Map</i> ketika <i>On Mission</i></p> <p>- Perbaiki halaman <i>Flight Plan</i></p>

	setting waypoint, serta menu  - Menu hanya tertera pada halaman <i>Flight Plan</i> - Tampilan setting waypoint <i>Flight Plan</i> dari horizontal menjadi <i>vertical</i> - Pengubahan <i>Icon Ardupilot</i> - Pengubahan fitur aplikasi hanya untuk <i>Fixed Wing</i> saja - Menu <i>Hide and Seek</i>	
2	- Menu <i>Simulation</i> dan <i>Help</i> di <i>Hide</i> (Tidak perlu dihapus)  - Pisah tampilan aplikasi menjadi dua bagian untuk dua monitor ( <i>Flight Data</i> dan <i>Flight Plan</i> )  - Halaman <i>Flight Data</i> terdiri dari <i>Quick Data 1</i> , <i>HUD</i> , <i>Quick Data 2</i> , <i>Connection Port</i> , <i>Map</i> , <i>Cam Overlay</i> , <i>Real Time UAV</i> .	- Perbaikan halaman <i>Flight Data</i>  - Perbaikan pemisahan aplikasi menjadi dua layar
3	--	--

#### 4.3.5 Hasil Perhitungan UEQ

Hasil dari *User Experience Questionnaire* adalah sebagai berikut:

Items																									
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
6	5	2	3	4	5	6	4	4	5	6	2	6	7	5	6	4	4	4	6	3	6	3	3	2	5
7	6	1	1	1	7	7	7	1	1	7	1	6	5	7	6	2	1	1	7	1	7	2	2	2	7
7	7	2	1	1	5	6	7	2	2	5	1	5	6	6	6	2	3	2	6	3	6	2	3	3	6
3	5	2	5	1	6	6	3	4	5	7	1	5	5	6	5	4	5	4	6	5	5	2	3	3	7
6	6	2	1	2	6	6	5	2	2	6	3	5	6	6	6	2	3	2	5	2	6	2	3	1	6

Gambar 11. Tabel Input Data Pengujian

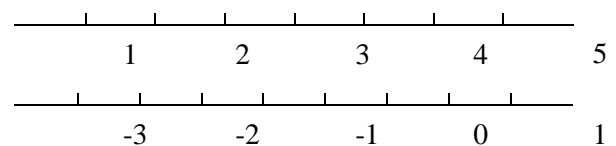
Dari 5 responden di dapat data seperti pada Gambar di atas, kuesioner yang diisi responden terdapat 26 pertanyaan dengan rentang nilai 1 sampai dengan 7. Sementara penilaian yang negatif dan positif masih di atur secara acak, sehingga angka yang tertera pada kuesioner baik tinggi maupun rendah tidak menunjukkan

penilaian yang semakin baik atau semakin buruk.

Items																									
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
2	1	2	1	0	1	2	0	0	-1	2	2	2	3	1	2	0	0	0	2	1	2	1	1	2	1
3	2	3	3	3	3	3	3	3	3	3	2	1	3	2	2	3	3	3	3	3	2	2	2	2	3
3	3	2	3	3	1	2	3	2	2	1	3	1	2	2	2	1	2	2	1	2	2	1	2	1	2
-1	1	2	-1	3	2	2	-1	0	-1	3	3	1	1	2	1	0	-1	0	2	-1	1	2	1	1	3
2	2	2	3	2	2	2	1	2	2	2	1	1	2	2	2	2	1	2	1	2	2	2	1	3	2

Gambar 12. Tabel Tranformasi Data

Kemudian nilai data 1 sampai dengan 7 diubah menjadi urutan -3 sampai 3 sesuai nilai garis bilangan sumbu X,



Maka didapat hasil seperti Gambar 12. seperti di atas. Agar memudahkan perhitungan UEQ data di kelompokkan berdasarkan 6 kategori penilaian. Sebaran pertanyaan secara acak sesuai tabel di bawah ini.

Tabel 19. Kategori Pengelompokan Pertanyaan

Skala Pengelompokan	Nomor pertanyaan
<i>Attractiveness</i>	1, 12, 14, 16, 24, 25
<i>Efficiency</i>	9, 20, 22, 23
<i>Perspicuity</i>	2, 4, 13, 21
<i>Dependability</i>	8, 11, 17, 19
<i>Stimulation</i>	5, 6, 7, 18
<i>Novelty</i>	3, 10, 15, 26

Item	Mean	Variance	Std. Dev.	No.	Left	Right	Scale
1	1.8	2.7	1.6	5	annoying	enjoyable	Attractiveness
2	1.8	0.7	0.8	5	not understandable	understandable	Perspicuity
3	2.2	0.2	0.4	5	creative	dull	Novelty
4	1.8	3.2	1.8	5	easy to learn	difficult to learn	Perspicuity
5	2.2	1.7	1.3	5	valuable	inferior	Stimulation
6	1.8	0.7	0.8	5	boring	exciting	Stimulation
7	2.2	0.2	0.4	5	not interesting	interesting	Stimulation
8	1.2	3.2	1.8	5	unpredictable	predictable	Dependability
9	1.4	1.8	1.3	5	fast	slow	Efficiency
10	1.0	3.5	1.9	5	inventive	conventional	Novelty
11	2.2	0.7	0.8	5	obstructive	supportive	Dependability
12	2.4	0.8	0.9	5	good	bad	Attractiveness
13	1.4	0.3	0.5	5	complicated	easy	Perspicuity
14	1.8	0.7	0.8	5	unlikable	pleasing	Attractiveness
15	2.0	0.5	0.7	5	usual	leading edge	Novelty
16	1.8	0.2	0.4	5	unpleasant	pleasant	Attractiveness
17	1.2	1.2	1.1	5	secure	not secure	Dependability
18	0.8	2.2	1.5	5	motivating	demotivating	Stimulation
19	1.4	1.8	1.3	5	meets expectations	does not meet expectations	Dependability
20	2.0	0.5	0.7	5	inefficient	efficient	Efficiency
21	1.2	2.2	1.5	5	clear	confusing	Perspicuity
22	2.0	0.5	0.7	5	impractical	practical	Efficiency
23	1.8	0.2	0.4	5	organized	cluttered	Efficiency
24	1.2	0.2	0.4	5	attractive	unattractive	Attractiveness
25	1.8	0.7	0.8	5	friendly	unfriendly	Attractiveness
26	2.2	0.7	0.8	5	conservative	innovative	Novelty

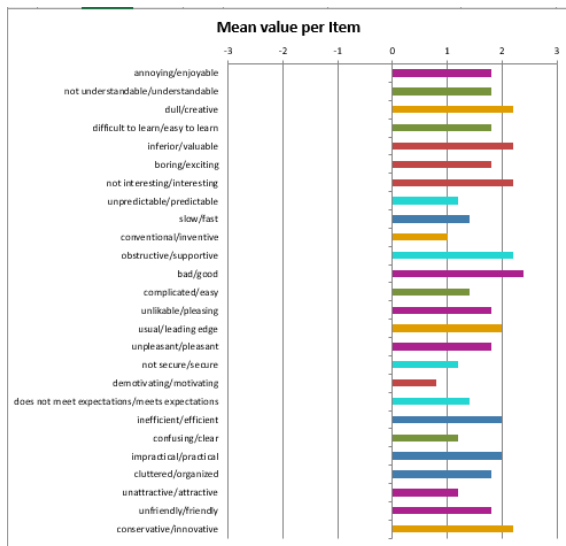
Gambar 13. Gambar Hasil Mean Data

Gambar 13. adalah hasil perhitungan nilai rata rata dan variasi dari setiap poin pertanyaan kuesioner.

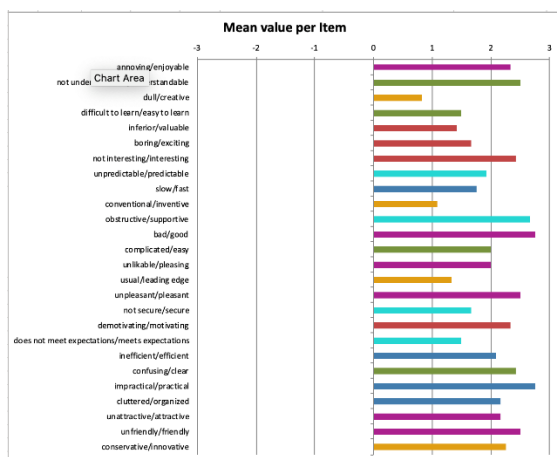
UEQ Scales (Mean and Variance)		
Attractiveness	1.800	0.21
Perspicuity	1.550	0.95
Efficiency	1.800	0.39
Dependability	1.500	0.97
Stimulation	1.750	0.66
Novelty	1.850	0.68

Gambar 14. Hasil *Mean* Tiap Kategori

Gambar 14. menunjukkan nilai hasil perhitungan rata-rata dan variasi dari setiap kategori penilaian. Rentang nilai adalah -3 sampai 3, artinya nilai 3 adalah nilai tertinggi secara keseluruhan sedangkan -3 penilaian dengan angka terburuk

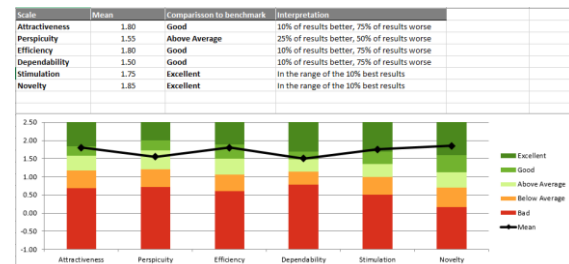
Gambar 15. Hasil *Mean* Tiap Poin

Gambar 15. adalah hasil perhitungan nilai rata-rata dan variasi dari setiap poin pertanyaan kuesioner



Gambar 16. Mean Masing-masing Pertanyaan

Pada Gambar 16. adalah nilai rata rata dari setiap poin pertanyaan. Dari rentang penilaian -3 sampai 3, semua poin pertanyaan berada di range positif, artinya impresi pengguna terhadap aplikasi yang diuji dapat disimpulkan cukup baik.

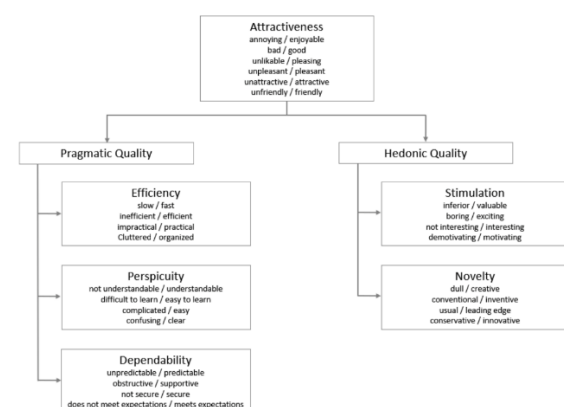


Gambar 17. Perbandingan Dengan Benchmark Data Set

Hasil perhitungan nilai *mean*/rata-rata dari setiap kategori kemudian dibandingkan dengan *Benchmark Data Set*. Pengelompokan kategori penilaian berdasarkan hasil rata rata setiap kategori adalah sebagai berikut:

- > 1,86 = *Excellent*
- > 1,6 = *Good*
- > 1,19 = *Above Average*
- > 0,7 = *Below Average*
- < 0,7 = *Bad*

Penilaian tertinggi ada di kategori *Attractiveness* dan penilaian terendah adalah *Novelty*. *Attractiveness* adalah penilaian utama (*goal-oriented*), jika penilaian pada kategori *Attractiveness* sudah baik berarti secara keseluruhan sistem sudah baik.



Gambar 18. Struktur Skala Penilaian UEQ [7]

*Perspicuity*, *Efficiency*, dan *Dependability* adalah aspek kualitas yang pragmatis (*goal-directed*), sedangkan

*Stimulation* dan *Novelty* adalah aspek kualitas hedonis (*not-goal-directed*) [7]. Sehingga jika disimpulkan secara garis besar *Attractiveness* adalah penilaian utama dari kategori lainnya.

## 5. KESIMPULAN

### 5.1 Kesimpulan

Berdasarkan hasil dan pembahasan yang didapatkan dari penelitian ini, terdapat beberapa kesimpulan, yakni sebagai berikut.

1. Pada penelitian ini, aplikasi *Mission Planner* dikembangkan dengan menggunakan bahasa C# .NET dan berjalan pada sistem operasi *Windows*. Aplikasi dapat menampilkan data penerbangan serta halaman untuk mengatur misi penerbangan. Pengguna yang telah melakukan koneksi aplikasi dengan *flight controller* dapat mengatur parameter penerbangan, mengatur misi, serta melihat data selama misi penerbangan.
2. Aplikasi ini dikembangkan menggunakan metode *prototype*, hasilnya pada setiap pengujian dan revisi yang didapatkan akan diterapkan pada aplikasi. Sehingga aplikasi yang dikembangkan akan lebih sesuai dengan keinginan pengguna.
3. Dari hasil pengujian *User Experience Quistionaire*, aplikasi *Mission Planner* ini mendapatkan *feedback* yang baik dari pengguna. Dengan nilai rata – rata untuk *Attractiveness* adalah 1,80 dari skala *minimum* -3,00 hingga skala *maximum* 3,00. Berarti aplikasi tersebut telah diterima dengan baik oleh pengguna.
4. Dari hasil perhitungan *Benchmark Data Set* pada pengujian UEQ aplikasi *Mission Planner*, dari ke-6 kategori, 2 kategori (*Stimulation* dan *Novelty*) mendapat hasil *Excellent*, 3 kategori (*Attractiveness*, *Efficiency*, *Dependability*) mendapat hasil *Good*, dan 1 kategori (*Perspiciuity*) mendapat *Above Average*. Artinya secara keseluruhan aplikasi dapat diterima oleh *user*, hanya saja butuh pengembangan lebih lanjut untuk kategori *Perspiciuity*.

### 5.2 Saran

Berdasarkan penelitian yang telah dilakukan, berikut merupakan saran untuk pengembangan selanjutnya:

1. Sesuai dengan kesimpulan nomor 4, hasil penilaian pengujian dengan

metode *User Experience Quistionaire* (UEQ) pada kategori *Perspiciuity* memiliki nilai terendah daripada yang lainnya, sehingga diharapkan untuk pengembangan lebih lanjut tampilan *User Interface* (UI) agar lebih jelas lagi.

2. Lebih konsisten dalam penggunaan versi aplikasi atau library untuk mengurangi tingkat kesalahan error saat menjalankan *Mission Planner* pada *Visual Studio*.
3. Disarankan untuk menambahkan fitur Camera overlay dan Real – time UAV untuk penelitian selanjutnya.

## UCAPAN TERIMA KASIH

Terima kasih kepada seluruh pihak-pihak terkait yang telah memberi bantuan baik dalam pengembangan sistem maupun dalam penulisan penelitian ini

## DAFTAR PUSTAKA

- [1] R. Hidayat and R. Mardiyanto, “Pengembangan Sistem Navigasi Otomatis Pada UAV (Unmanned Aerial Vehicle) dengan GPS (Global Positioning System) Waypoint,” J. Tek. ITS, vol. 5, no. 2, 2017, doi: 10.12962/j23373539.v5i2.16342.
- [2] Aris Susilo, “Pengembangan Sistem *Autopilot* Pada Wahana Udara Tanpa Awak *Fixed Wing* Proto-03”, Tugas Akhir, 2015
- [3] A. M. Handayani and I. N. Rifa’i, “Sistem Ground Control Station Berbasis Mobile Untuk,” Semin. Nas. Teknol. Terap., vol. 2, no. 16, pp. 23–30, 2018.
- [4] Y. Hong, J. Fang, and Y. Tao, “Ground control station development for autonomous UAV,” Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 5315 LNAI, no. PART 2, pp. 36–44, 2008, doi: 10.1007/978-3-540-88518-4\_5.
- [5] A. M. Handayani and B. Sumanto, “Sistem Ground Control Station untuk Pengamatan dan Pengendalian Unmanned Aerial Vehicle,” Semin. Nas. Teknol. Terap., vol. 2, pp. 1000–1003, 2016.
- [6] Ganney, P. S., Pisharody, S., & Claridge, E. (2013). Software Engineering. In Clinical Engineering: A Handbook for Clinical and Biomedical Engineers.

<https://doi.org/10.1016/B978-0-12-396961-3.00009-3>

- [7] M. Schrepp, "User Experience Questionnaire Handbook Version 8," in URL:  
[https://www.researchgate.net/publication/303880829\\_User\\_Experience\\_Questionnaire\\_Handbook\\_Version\\_2](https://www.researchgate.net/publication/303880829_User_Experience_Questionnaire_Handbook_Version_2). (Accessed: 02.02.2017), 2019, pp. 1–15.