

# Pengembangan Sistem Informasi Pemantauan Pengguna PUTA Berbasis Android dengan Menggunakan Metode *Geofencing*

Muhammad Bagus Istighfar, Daniel Oranova Siahaan, dan Muhammad Riduwan  
Departemen Teknik Informatika, Institut Teknologi Sepuluh Nopember (ITS)  
*e-mail*: daniel@if.its.ac.id

**Abstrak**—Perkembangan Pesawat Udara Tanpa Awak (PUTA) seperti *drone* di Indonesia semakin meningkat dan banyak digunakan di berbagai sektor seperti pertanian, kehutanan, serta logistik. Namun, masih banyak pengguna PUTA yang menerbangkan tanpa memperhatikan regulasi yang ada. Oleh karena itu, dibutuhkan sistem informasi terkait pemantauan penerapan regulasi penerbangan PUTA bagi pengguna berupa *Unmanned Aerial Vehicle* ITS (UAVITS). Sistem informasi UAVITS merupakan sistem yang dapat memantau penerbangan PUTA secara *real-time*. Adapun metode yang digunakan dalam pengembangan sistem informasi pemantauan pengguna PUTA berbasis *android* pada penelitian ini yaitu metode *geofencing*. Metode *geofencing* dapat digunakan sebagai acuan penentuan wilayah penerbangan sesuai dengan ketentuan yang ada serta mengontrol pergerakan PUTA agar tidak melanggar batas wilayah penerbangan yang diizinkan. Penggunaan metode ini dapat memaksimalkan pengimplementasian regulasi penerbangan yang berlaku di Indonesia. Hasil dari penelitian ini menunjukkan Sistem UAVITS Apps berbasis Android berhasil diimplementasikan sesuai dengan perancangan yang dilakukan. Sistem berhasil diuji melalui tiga pengujian yaitu pengujian pada fitur *monitoring*, pengujian pada fitur *check-zone*, dan pengujian keseluruhan sistem menggunakan *User Acceptance Test* (UAT). Penelitian ini dapat menjadi acuan untuk mendukung penerapan regulasi penerbangan PUTA atau *drone* bagi pengguna serta membantu AirNav dalam melakukan proses pemantauan pergerakan PUTA sesuai regulasi yang berlaku. Pengembangan sistem UAVITS Apps berbasis Android ke depannya diharapkan dapat menggunakan simbol *drone* yang tepat pada fitur *monitoring*, agar tidak perlu mengatur arah hadap simbol sesuai dengan pergerakan PUTA.

**Kata Kunci**—*Android*, *Geofencing*, Pemantauan Pengguna, PUTA, *User Acceptance Test*.

## I. PENDAHULUAN

SAAT ini perkembangan Pesawat Udara Tanpa Awak (PUTA) seperti *drone* di Indonesia semakin meningkat. Pada awalnya PUTA hanya dikenal sebagai alat perekaman melalui udara. Kini, PUTA banyak digunakan di berbagai sektor seperti pertanian, kehutanan, logistik dan lainnya. Namun, masih banyak pengguna yang menerbangkan PUTA tanpa memperhatikan regulasi yang ada. Hal ini membuat pengawasan terhadap pelaksanaan regulasi penerbangan *drone* tidak terkontrol dengan baik, sehingga akan menimbulkan beberapa permasalahan terkait keamanan wilayah penerbangan dan lalu lintas udara. Selain berpotensi mengancam keamanan dan keselamatan lalu lintas pesawat udara berawak, penggunaan PUTA melanggar ketentuan yang berlaku juga dapat mengakibatkan risiko lainnya di darat misalnya kerugian yang dapat menimpa orang dan benda di bawahnya [1].

Salah satu sistem informasi terkait pemantauan penerapan

regulasi penerbangan *drone* yang berkembang yaitu *Unmanned Aerial Vehicle* ITS (UAVITS). Sistem informasi UAVITS merupakan sistem yang dapat memantau penerbangan PUTA secara *real-time*. UAVITS telah memiliki sistem bagi admin dan pengguna berbasis website. UAVITS dapat membantu AirNav Surabaya selaku instansi yang berwenang dalam mengawasi penerbangan PUTA di area wewenang AirNav Surabaya. Selain itu, UAVITS dapat mempermudah pengguna PUTA dalam penerapan regulasi penerbangan PUTA. Sehingga sistem informasi ini dapat menjadi gebrakan awal yang baik untuk pengendalian penerbangan PUTA ke depannya. Namun, sistem tersebut memiliki beberapa kekurangan seperti peta yang ada masih belum sesuai dengan standar peta lalu lintas udara. Sistem ini telah menerapkan metode *geofencing*, namun metode yang ada masih belum sesuai dengan regulasi yang ada. Selain itu, sistem ini masih berbasis *website* yang tidak responsif terhadap berbagai ukuran perangkat seperti *mobile*.

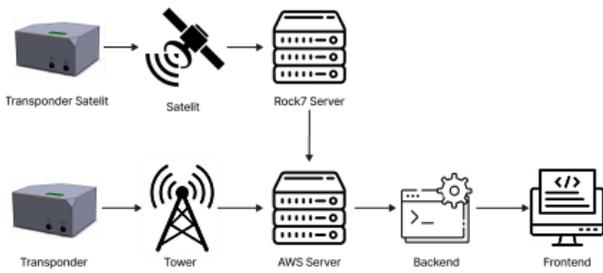
Pengembangan sistem informasi UAVITS di sisi *Android mobile* dapat memperluas penggunaan sistem. *Android mobile* memungkinkan pengguna untuk memakai sistem dengan lebih efektif dan efisien karena dapat diakses di mana saja. Sistem informasi UAVITS di sisi *Android mobile* ini menggunakan *geofencing*. Metode *geofencing* mengontrol pergerakan PUTA agar tidak melanggar batas wilayah penerbangan yang diizinkan. Selain itu metode *geofencing* juga digunakan sebagai acuan penentuan wilayah penerbangan sesuai dengan ketentuan yang ada. Penerapan metode *geofencing* ini dapat memaksimalkan pengimplementasian regulasi penerbangan yang berlaku di Indonesia.

Untuk menjawab permasalahan mengenai pemantauan regulasi penerbangan terhadap penggunaan PUTA di batas wilayah yang diizinkan dapat dikembangkan sistem informasi UAVITS bagi pengguna berbasis *Android mobile* dengan menggunakan metode *geofencing* untuk pemetaan batas wilayah penerbangan sesuai regulasi dan pengendalian penerbangan PUTA atau *drone*. Sehingga, dengan adanya penelitian ini dapat menjadi acuan untuk mendukung penerapan regulasi penerbangan PUTA atau *drone* bagi pengguna serta membantu AirNav dalam melakukan proses pemantauan pergerakan PUTA sesuai regulasi yang berlaku.

## II. TINJAUAN PUSTAKA

### A. *Transponder*

*Transponder* merupakan suatu komunikasi antara pengguna dengan PUTA yang mampu mengirimkan posisi PUTA agar ditampilkan dalam sistem kontrol [2]. Skema pengiriman data oleh *transponder* pada penelitian ini dibagi menjadi



Gambar 1. Skema pengiriman data transponder GSM dan satelit.

dua jenis, yaitu *transponder* GSM dan *transponder* satelit seperti pada Gambar 1.

**B. Regulasi Penerbangan PUTA**

Peraturan Menteri Perhubungan Nomor 37 tahun 2020 menjelaskan kawasan mana saja PUTA tidak boleh diterbangkan. Dalam peraturan tersebut kawasan-kawasandiatur dikategorikan menjadi lima bagian, yaitu: (1) Kawasan udara terlarang merupakan area militer, seperti area sumber energi, kawasan hutan lindung, dan juga Zona Ekonomi Eksklusif (ZEE). (2) Kawasan Udara Terbatas merupakan Kawasan yang meliputi rumah susun, apartemen, pergudangan, atau kepemilikan individual dan bersama. (3) Kawasan keselamatan operasi penerbangan (KKOP) suatu bandara-udara. Kawasan berada di dalam radius 3 *Nautical Mile* dari titik koordinat helipad yang berlokasi di luar KKOP suatu Bandar udara. (4) *Controlled airspace* berupa pemanduan lalu lintas penerbangan, pelayanan informasi penerbangan, dan dana pelayanan kesiagaan. (5) *Uncontroller air space* berupa pelayanan informasi penerbangan, pelayanan dan pelayanan saran lalu lintas penerbangan.

**C. Geofencing**

*Geofencing* adalah teknologi yang digunakan untuk memantau objek bergerak dengan menggunakan GPS. Fungsi utama *geofencing* yaitu untuk melakukan pemantauan jarak jauh suatu objek dari peta virtual ketika objek keluar atau memasuki daerah yang dibatasi *geofence* (pagar virtual) [3].

**D. Alat Pengembangan Sistem**

**1) Android Studio**

Android studio adalah *open source Integrated Development Enviroment* (IDE) untuk pengembangan aplikasi Android [4].

**2) ExpressJs**

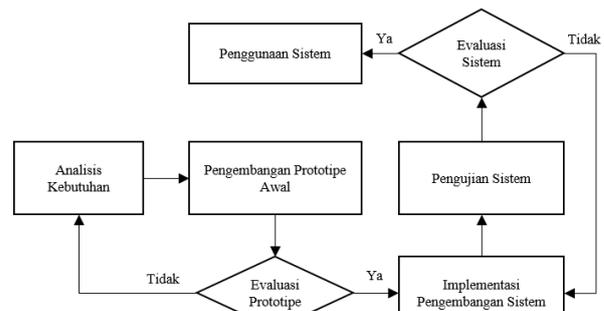
*ExpressJs* adalah sebuah kerangka kerja yang terdapat dalam NodeJs untuk pengembangan aplikasi web, 14 *service API*, *routing*, dan *security* [5].

**3) MongoDB**

*MongoDB* adalah basis data NoSQL yang bersifat *document-based*. Data yang disimpan dalam basis data MongoDB berupa *file JSON* yang disebut dengan istilah BSON (Binary JSON). Model datanya berbasis dokumen membuat pengguna tidak perlu merancang struktur tabel seperti pada SQL [6].

**E. Model Prototipe**

Metode prototipe merupakan metode pengembangan perangkat lunak berupa model fisik kerja sistem dan berfungsi sebagai versi awal dari sistem [7]. Prototipe tersebut akan



Gambar 2. Tahapan SDLC model prototipe.

dievaluasi oleh pengguna dan dipakai untuk menyaring kebutuhan pengembangan perangkat lunak. Rancangan aplikasi awal mulanya berbentuk *mockup* prototipe. Kemudian prototipe dievaluasi pengguna dan menjadi bahan rujukan bagi pengembang *software* untuk merancang sistem [8]. Tahapan SDLC dengan model prototipe terdapat pada Gambar 2.

**III. DESAIN DAN PERANCANGAN SISTEM**

**A. Analisis Kebutuhan**

Tahapan kebutuhan merupakan tahapan untuk penggalan kebutuhan yang nantinya akan digunakan untuk pengembangan sistem. Tahapan analisis kebutuhan bertujuan untuk menggali kebutuhan *stakeholders* yang dibutuhkan oleh sistem. Tahapan ini menghasilkan daftar kebutuhan fungsional dan non-fungsional yang menjadi acuan dalam pembuatan diagram kasus penggunaan, spesifikasi kasus penggunaan, diagram aktivitas, diagram hubungan entitas pada dokumen Spesifikasi Kebutuhan Perangkat Lunak (SKPL). *Stakeholders* yang berkaitan dengan kebutuhan sistem ini meliputi *Beehive Drones* selaku pemilik perusahaan PUTA dan *transponder*, serta *AirNav Surabaya* selaku pihak yang mengawasi lalu lintas udara [9].

**B. Pengembangan Prototipe Awal**

Setelah mendapatkan kebutuhan dari pengguna dan *stakeholders* terkait, daftar kebutuhan tersebut dikembangkan prototipe awal. Prototipe awal yang dikembangkan berfokus pada alur penggunaan sistem oleh pengguna dengan menggunakan aplikasi Figma.

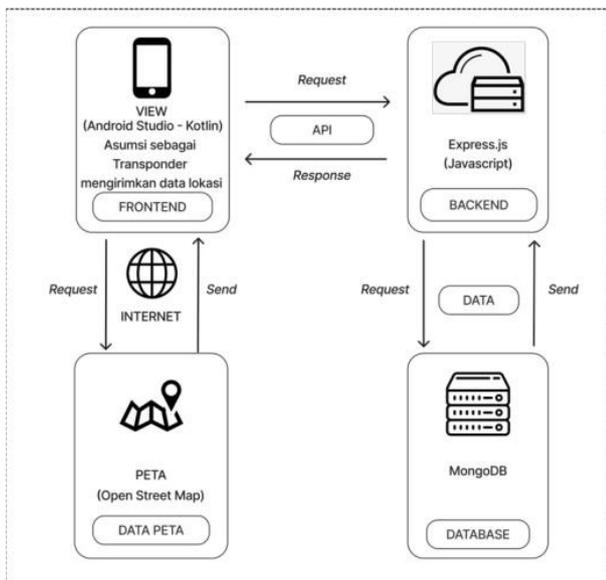
**C. Evaluasi Prototipe**

Prototipe awal yang dikembangkan akan dipresentasikan kepada pengguna dan *stakeholders* terkait untuk mendapatkan *feedback*. Jika evaluasi telah sesuai, maka dilanjutkan ke tahap implementasi pengembangan sistem. Jika tidak, maka akan kembali diulang tahap analisis kebutuhan, perbaikan prototipe, dan evaluasi prototipe.

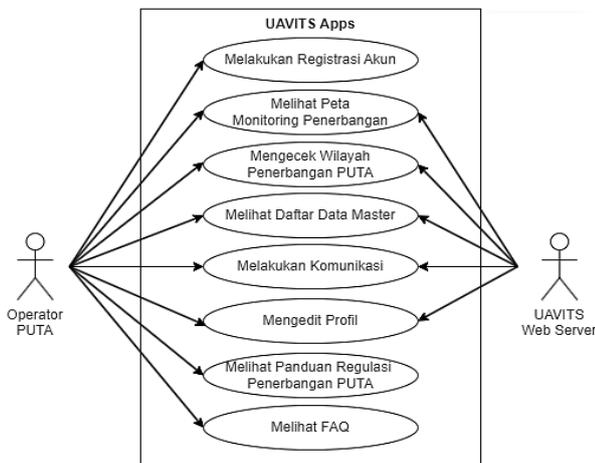
**D. Implementasi Pengembangan Sistem**

Prototipe awal yang telah disetujui oleh pengguna dan *stakeholders* terkait akan dilanjutkan ke proses pengembangan sistem informasi. Rancangan arsitektur sistem yang digunakan terdapat pada Gambar 3.

Pengembang akan mengembangkan sistem di sisi *frontend* dan *backend*. Pengembangan sistem informasi ini menggunakan *requirement*, seperti Bahasa pemrograman *frontend*: Kotlin, Bahasa pemrograman *backend*: Javascript, *Integrated Development Environment* (IDE): Android Studio, *Framework backend*: Express.js, dan Open Street Map.



Gambar 3. Rancangan arsitektur sistem.



Gambar 4. Diagram kasus penggunaan UAVITS apps.

E. Pengujian Sistem

Pengujian sistem menggunakan metode *User Acceptance Test (UAT)* yang dilakukan oleh perwakilan dari *stakeholders*. Penguji mengikuti dokumen skenario pengujian yang didasarkan pada dokumen SKPL yang disepakati bersama.

F. Evaluasi Sistem

Evaluasi sistem merupakan tahapan di mana *stakeholders* akan menentukan apakah sistem dapat sudah sesuai dengan harapan atau tidak. Jika tidak maka proses implementasi pengembangan kembali dilakukan.

IV. HASIL DAN PEMBAHASAN

A. Analisis Kebutuhan

Analisis kebutuhan dilakukan dengan teknik wawancara pada *stakeholders* terkait menghasilkan diagram kasus penggunaan yang digunakan sebagai acuan untuk mengembangkan sistem UAVITS Apps yang divisualisasikan pada Gambar 4 dan dijabarkan pada Tabel 1.

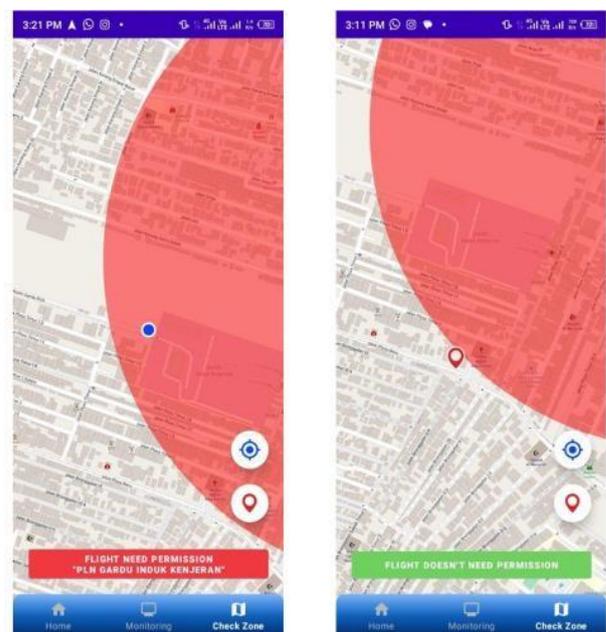
B. Pengembangan Prototipe Awal

Berdasarkan Tabel 1 dibuat beberapa daftar halaman yang

Tabel 1.

Penjabaran Diagram Kasus Penggunaan UAVITS

ID	Kasus Penggunaan Utama	Deskripsi Kasus Penggunaan Utama
KU01	Melakukan Registrasi Akun	Operator PUTA melakukan registrasi akun pada sistem
KU02	Melihat Peta Penerbangan	Operator PUTA <i>memonitoring</i> penerbangan PUTA secara <i>realtime</i> pada peta
KU03	Mengecek Wilayah Penerbangan PUTA	Operator PUTA mengecek status izin wilayah penerbangan PUTA
KU04	Melihat Daftar Data Master	Operator PUTA mengakses dan melihat data master yang tersimpan dalam <i>database</i>
KU05	Melakukan Komunikasi	Operator PUTA dapat melakukan komunikasi dengan admin dari AirNav terkait koordinasi sebelum penerbangan
KU06	Mengedit Profil	Operator PUTA dapat mengedit data profil operator
KU07	Melihat Panduan Regulasi Penerbangan PUTA	Operator PUTA dapat melihat panduan sebagai operator dalam mengikuti dan menerapkan regulasi penerbangan PUTA yang berlaku
KU08	Melihat FAQ	Operator PUTA dapat melihat Forum <i>Answer and Question (FAQ)</i> untuk membantu dalam penggunaan aplikasi



Gambar 5. Fitur *checkzone* menggunakan lokasi sekarang dan lokasi pilihan (kanan).

menjadi acuan dalam pengembangan prototipe. Jumlah halaman yang digunakan dalam sistem ini sebanyak 40 halaman. Langkah selanjutnya adalah melakukan proses pengumpulan ide atau *moodboard*. Kemudian ide-ide tersebut dikembangkan menjadi desain antar muka sistem seperti desain antarmuka sistem. Setelah membuat antarmuka sistem secara keseluruhan, setiap antarmuka saling dihubungkan membentuk skenario alur kejadian dari awal sistem dimulai sesuai dengan diagram aktivitas yang ada. Setiap antarmuka dapat saling terhubung dan berinteraksi melalui komponen antarmuka sistem seperti tombol. Sebagai contoh tampilan prototipe pada fitur *checkzone* ditunjukkan pada Gambar 5.

C. Evaluasi Prototipe

Tahap ini dilakukan sebanyak dua kali dengan *Beehive Drones* dan AirNav Surabaya. *Beehive Drones* menyatakan tidak perlu ada yang dievaluasi, semuanya sudah sesuai.

Tabel 2.  
Hasil Evaluasi Prototipe

No	Evaluasi
1	Perubahan nama <i>checklog</i> menjadi <i>checklist</i> pada menu " <i>FlightList</i> "
2	Revisi kesalahan tulisan pada header " <i>Flight List</i> " pada menu " <i>Flight List</i> "
3	Perubahan kata-kata " <i>License</i> " menjadi " <i>Permit</i> " pada menu " <i>Flight List</i> " dan " <i>Post Flight Report</i> "
4	Kata " <i>DNP</i> " tidak boleh disingkat, harus dijabarkan menjadi " <i>Directorate of Aviation Navigation</i> " pada menu " <i>Flight List</i> "
5	Perubahan kata " <i>Origin</i> " menjadi " <i>Take Off Point</i> " pada menu " <i>Flight List</i> "
6	Perubahan kata " <i>Destination</i> " menjadi " <i>Landing Point</i> " pada menu " <i>Flight List</i> "
7	Kata " <i>Notam</i> " tidak boleh disingkat, harus dijabarkan menjadi " <i>Notice to Airmen</i> " pada menu " <i>Post Flight Report</i> "
8	Pilihan " <i>Serious Injury</i> " pada " <i>Flight Accident</i> " dihapus dan pilihan " <i>Fatal Injury</i> " diganti dengan " <i>Human Fatal Damage</i> " pada menu " <i>Post Flight Report</i> "
9	Fitur <i>Checkzone</i> tidak perlu meliputi daerah seperti faktorkerapatan bangunan, cukup kawasan militer, Objek Vital Nasional, dan KKOP. Karena AirNav tidak pernah langsung melihat bagaimana kondisi asli permukaan di wilayah tersebut.

Tabel 3.  
Hasil Evaluasi Prototipe

No	Longitude	Latitude	Altitude (ft)	Ground Speed (knots)	Status	Pesan
1	112.79638	-7.27711	109.908	0.00000	safe	Safe flight
2	112.79642	-7.27656	0.00000	2.40555	warning	0.0327 nmi until exiting the flight zone
3	112.79635	-7.27663	0.00000	1.32046	warning	0.0368 nmi until exiting the flight zone
...	...	...	...	...	...	...
137	112.79634	-7.27669	0.00000	0.76392	warning	0.0403 nmi until exiting the flight zone
138	112.79648	-7.27688	31.2153	4.72640	warning	0.0523 nmi until exiting the flight zone
139	112.79647	-7.27691	0.00000	5.83637	safe	Safe flight
140	112.79647	-7.27694	0.00000	6.96111	safe	Safe flight
...	...	...	...	...	...	...
227	112.79417	-7.27617	75.3473	12.6438	danger	Out from the flight zone
228	112.79419	-7.27615	75.3473	10.3598	danger	Out from the flight zone
...	...	...	...	...	...	...
399	112.79712	-7.27810	1359.48	0.00000	safe	The flight exceeds the maximum altitude limit
...	...	...	...	...	...	...

Total Data/Waktu Terbang = 754 / 12 menit 50 detik

Namun ada beberapa catatan pada saat evaluasi prototipe dengan AirNav Surabaya. Adapun evaluasi yang disampaikan tertera pada Tabel 2.

D. Implementasi Pengembangan Sistem

Pengembangan sistem UAVITS Apps menggunakan metode *geofencing* untuk menentukan status kawasan apakah membutuhkan perizinan atau tidak berdasarkan lokasi yang ditentukan. Data yang digunakan yaitu data wilayah Kota Surabaya dan Sidoarjo yang terdapat di AIM Indonesia hanya data lokasi bandara. Data militer di dapat dari penggalian informasi melalui internet. Sedangkan data objek vital nasional didapat dari situs sistem.obvit.jatim.polri.go.id. Data daerah yang membutuhkan perizinan pada sistem UAVITS Apps sebanyak 52 data yang terdiri dari 35 data daerah militer, 16 data daerah objek vital nasional, dan 1 daerah KKOP.

Daerah yang membutuhkan perizinan ditandai dengan *geofence* berbentuk lingkaran berwarna merah dengan radius sebesar 500 m untuk daerah militer dan objek vital nasional. Sedangkan untuk daerah KKOP menggunakan radius 4000 m yang merupakan jarak dari titik tengah KKOP ke batas daerah pacu landas. Namun, penentuan radius sebesar 500 m untuk daerah militer dan objek vital nasional bukan merupakan ukuran yang sesuai dengan keadaan sebenarnya karena tidak ada regulasi tetap yang mengatur. Setiap daerah memiliki luas daerah yang berbeda. Sehingga radius untuk daerah militer dan objek vital nasional seharusnya disesuaikan dengan luas daerah masing-masing. Selain itu penentuan radius untuk setiap KKOP juga tidak bisa disamakan, tergantung penetapan radius KKOP yang ditetapkan oleh Direktur Jenderal seperti pada PM 37 tahun 2020 poin 2.2.3. Penentuan radius daerah yang membutuhkan perizinan pada fitur *checkzone* ini merupakan rekomendasi dari AirNav

Surabaya sebagai data sementara untuk pengujian.

E. Pengujian Sistem

1) Pengujian Fitur Monitoring

Pengujian fitur *monitoring* dilakukan dengan cara menerbangkan *drone*. Dalam pengujian ini digunakan ponsel pintar sebagai pengganti *transponder* dalam mengirimkan data lokasi. Ponsel ditempelkan pada *drone* untuk dapat mengirimkan data lokasi sesuai pergerakan *drone*. Pengujian ini menghasilkan empat variabel yang dikirimkan oleh ponsel pada server, yaitu *longitude*, *latitude*, *altitude* dalam satuan ft, dan *ground speed* dalam satuan knot. Hasil pengujian fitur *monitoring* disajikan pada Tabel 3.

Pada Tabel 3 terdapat tiga tipe status penerbangan yakni aman (*safe*), peringatan (*warning*), dan berbahaya (*danger*) dengan masing-masing pesan (*message*). Data yang didapatkan selama 12 menit 50 detik sebanyak 754 data. Data 1 menunjukkan *altitude* dengan nilai 109.908. Sedangkan Data 2 menunjukkan *altitude* dengan nilai 0. Hal ini dikarenakan pada algoritma *altitude* data pertama digunakan sebagai acuan dari permukaan laut hingga permukaan daratan saat *drone* belum memiliki ketinggian. Sehingga *altitude* data selanjutnya selalu dikurangi dengan *altitude* Data 1 untuk menghasilkan *altitude* yang diharapkan bukan diukur dari atas permukaan laut. Pada Data 2, 3, 137, dan 138 menunjukkan *drone* memasuki daerah peringatan, 100 m dari tepi batas daerah perizinan. Jarak *drone* pada tepi batas daerah perizinan yang terdapat pada kolom pesan memiliki satuan *Nautical Miles*. Data 227 dan 228 menunjukkan *drone* memasuki daerah berbahaya atau di luar daerah yang diizinkan. Sedangkan, pada data 399 menunjukkan *drone* memiliki *altitude* yang melebihi 120 m atau sekitar 393 ft. Namun, dalam pengujian ini menghasilkan nilai *altitude* yang tidak sesuai dikarenakan ponsel yang digunakan tidak memiliki

Tabel 4.  
Hasil Pengujian Fitur *Checkzone*

No	Nama	Longitude	Latitude	Status	Lokasi Sekarang	Lokasi Pilihan	Keterangan
1	Bundaran ITS	112.790	-7.278	Safe	Valid	Valid	-
2	Expom Kodim 0831 Mulyorejo	112.781	-7.260	Danger	Valid	Valid	-
3	Jalan Ploso Baru	112.771	-7.256	Safe	Valid	Valid	-
4	PLN Gardu Induk Kenjeran	112.770	-7.254	Danger	Valid	Valid	-
5	Koramil 0831/02 Tambaksari	112.755	-7.257	Danger	Valid	Valid	-
...	...	...	...	...	...	...	...
25	PT. Sinar Mas Agro Resources	112.758	-7.326	Danger	TidakValid	TidakValid	Tempat sudah pindah
Total Valid = 24; Total Safe = 7; Total Tidak Valid = 1; Total Danger = 18							

Tabel 5.  
Hasil Pengujian UAT Pada Sistem UAVITS

Penguji	Login	Register	Flight	Drone	Pilot	Post Flight Report	Flight History	Contact	Regulation	FAQ	Profile
1	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
2	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
3	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
4	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
5	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
6	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Jumlah Kasus Pengujian Selesai	2	1	8	6	6	6	2	2	2	2	2
Tingkat Penyelesaian	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%

sensor barometer. Sehingga, nilai *altitude* yang didapatkan tidak akurat dan tidak stabil. Nilai *altitude* yang didapatkan tidak dapat menggunakan *Global Positioning System* (GPS).

Untuk hasil tampilan pengujian fitur *monitoring* dapat dilihat pada Gambar 6. Pada Gambar 6 ditunjukkan saat *drone* berada pada daerah aman, kemudian memasuki daerah peringatan mulai dari 100 m ke batas tepi daerah perizinan, dan kemudian keluar menuju daerah berbahaya. Pada saat *drone* berada pada daerah peringatan, muncul pesan peringatan dengan jarak satuan *Nauticle Miles*. Pada saat *drone* memasuki daerah berbahaya, ponsel akan menampilkan notifikasi secara terus menerus untuk memperingatkan pengguna agar segera kembali ke daerah yang diizinkan. Selain itu ditunjukkan pula bahwa sistem belum memiliki fitur untuk mengarahkan arah hadap simbol *drone* dengan sesuai.

### 2) Pengujian Fitur *Checkzone*

Dari 50 titik daerah yang memerlukan perizinan untuk terbang, diambil titik sampel secara acak sebanyak 25 titik baik di dalam daerah yang membutuhkan perizinan untuk terbang ataupun tidak untuk dilakukan pengujian fitur *checkzone*. Setiap percobaan pengujian dilakukan dengan dua cara yaitu menggunakan lokasi sekarang dan menggunakan lokasi pilihan. Hasil pengujian fitur *checkzone* disajikan pada Tabel 4.

Pada Tabel 4, pengujian memiliki nilai valid yang menyatakan hasil telah sesuai dengan lokasi yang sebenarnya, dan sebaliknya. Jika tidak valid, maka pengujian yang dilakukan pada titik sampel tersebut tidak sesuai dengan lokasi yang sebenarnya. Pada Tabel 4 ditunjukkan terdapat 24 data valid dari 25 data. 1 data tidak valid dikarenakan tempat sudah pindah. Hal ini dikarenakan pengambilan data hanya melalui Google Maps bukan data lokasi terbaru yang sebenarnya. Pengujian dilakukan di 7 daerah aman (*safe*) atau daerah yang tidak membutuhkan perizinan untuk terbang dan 18 daerah

berbahaya (*danger*) atau daerah yang membutuhkan perizinan untuk terbang.

### 3) Pengujian User Acceptance Test (UAT)

UAT dilakukan pada enam pengujian yang terdiri dari tiga AirNav Surabaya dan tiga dari *Beehive Drones*. UAT tidak dilakukan secara mandiri dengan mengikuti panduan yang ada di dokumen pengujian. Pengujian memiliki 11 modul dengan 39 kasus pengujian yang diuji oleh 6 pengujian. Hasil dari pengujian UAT ini adalah seluruh kasus pengujian (*test case*) 100% memiliki status “Berhasil” disajikan pada Tabel 5 Hasil Pengujian UAT pada Sistem UAVITS Apps.

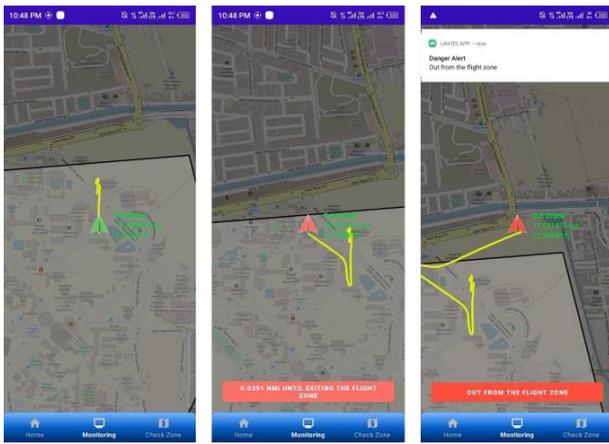
### F. Evaluasi Sistem

Setelah melalui proses pengujian, *stakeholders* menentukan bahwa sistem UAVITS Apps telah sesuai dengan harapan dan memenuhi kebutuhan sistem yang dibutuhkan. Proyek pengembangan Aplikasi *Android* UAVITS Apps bagi pengguna telah lolos pengujian dan evaluasi sistem.

### G. Pembahasan

Pengembangan sistem UAVITS Apps menggunakan model prototipe. Penggunaan model prototipe dalam pengembangan ini sangat berhubungan langsung kepada pengguna dan *stakeholders* dengan harapan mendapatkan pendapat langsung yang sesuai dengan keinginan dan kebutuhan pengguna dan *stakeholders* terkait yang akan berguna untuk pengembangan sistem. Pemilihan model ini akan memudahkan dalam proses pengembangan sistem selanjutnya, mendeteksi kesalahan lebih awal, serta mendapatkan *feedback* dari pengguna dan *stakeholders* terkait untuk mencapai tujuan sistem secara maksimal.

Pada pengujian fitur *monitoring* menunjukkan bahwa fitur *monitoring* telah berjalan dengan baik dan sesuai yang melibatkan tiga zona penerbangan terhadap daerah yang diizinkan yaitu zona aman, zona peringatan, dan zona berbahaya. Pengujian fitur *monitoring* berjalan selama 12



Gambar 6. Tampilan hasil pengujian fitur *monitoring* pada daerah aman (kiri), daerah peringatan (tengah), dan daerah bahaya (kanan).

menit 50 detik yang menghasilkan 754 data lokasi. Namun, untuk variabel *altitude*, pengujian belum bisa dinyatakan akurat. Hal ini dikarenakan ponsel yang digunakan untuk pengujian tidak memiliki sensor barometer.

Kemudian, pada pengujian fitur *checkzone* yang menggunakan 52 titik daerah yang membutuhkan perizinan menghasilkan hasil yang cukup sesuai. Dari 52 titik daerah yang diujikan menggunakan cara menentukan lokasi berdasarkan lokasi sekarang maupun lokasi pilihan ada 1 titik yang tidak valid. Hal ini dikarenakan titik tersebut tidak sesuai dengan lokasi yang sebenarnya. Ini menunjukkan bahwasanya data titik yang digunakan tidak bisa dikatakan akurat. Data yang dikumpulkan terdiri dari data kawasan militer, data kawasan objek vital nasional dan data KKOP. Data yang digunakan seperti data militer dan data objek vital nasional tidak dari data yang terdaftar di AIM Indonesia. AIM Indonesia merupakan sebuah situs yang menyimpan data yang berkaitan dengan lalu lintas udara, termasuk kawasan *restricted* dan *prohibited* di Indonesia. Data wilayah Kota Surabaya dan Sidoarjo yang terdapat di AIM Indonesia hanya data lokasi bandara atau KKOP. Data militer di dapat dari penggalian informasi melalui internet. Sedangkan data objek vital nasional didapat dari situs [sistem.obvit.jatim.polri.go.id](http://sistem.obvit.jatim.polri.go.id).

Kemudian, berdasarkan hasil pengujian UAT yang dilakukan, keenam penguji dari *stakeholders* berhasil menyelesaikan 11 modul pengujian yang terdiri dari 39 kasus pengujian dengan tingkat persentase keberhasilan 100%. Penguji menyatakan sistem telah berjalan dengan baik dan sesuai dengan keinginan serta kebutuhan yang disepakati saat proses analisis kebutuhan.

Berdasarkan hasil setiap pengujian, sistem UAVITS Apps dapat dikatakan layak digunakan pengguna. Hal ini diperkuat dengan kelebihan yang dimiliki sistem UAVITS Apps jika dibandingkan dengan sistem yang menjadi dasar pengembangan sistem ini, yaitu Sydrone Map. Aplikasi UAVITS Apps memiliki fitur-fitur yang mencakup seluruh kebutuhan AirNav Surabaya dalam melakukan proses pemantauan pergerakan PUTA. Selain itu, sistem UAVITS Apps tidak menggunakan metode SMCE dalam memetakan kawasan yang membutuhkan perizinan untuk terbang seperti halnya pada sistem Sydrone Map, sehingga memiliki tingkat keakuratan yang lebih tinggi. Pada sistem Sydrone Map memiliki keakuratan data sebesar 62.5%, sedangkan pada

sistem UAVITS Apps hanya memiliki satu data tidak valid pada fitur *checkzone*.

## V. KESIMPULAN/RINGKASAN

Sistem UAVITS Apps berbasis android berhasil diimplementasikan sesuai dengan perancangan yang dilakukan kesimpulan yang didapatkan penulis, bahwa: (1) Sistem dapat membantu AirNav Surabaya dalam melakukan proses pemantauan pergerakan PUTA sesuai dengan menerapkan regulasi yang berlaku. (2) Sistem UAVITS Apps dikembangkan dengan model prototipe yang sangat berhubungan langsung kepada pengguna dan *stakeholders* dengan harapan mendapatkan pendapat langsung yang sesuai dengan keinginan dan kebutuhan pengguna dan *stakeholders* terkait yang akan berguna untuk pengembangan sistem. (3) Sistem UAVITS Apps menggunakan metode *geofencing* untuk mengetahui apakah PUTA berada pada daerah yang diizinkan untuk terbang atau tidak. Selain itu, metode *geofencing* juga digunakan untuk membantu pengguna dalam menentukan kawasan yang membutuhkan perizinan untuk terbang atau tidak. (4) Sistem dibangun menggunakan kerangka kerja ExpressJs yang menggunakan basis data NoSQL MongoDB untuk implementasi pengembangan di sisi server dan Android Studio untuk implementasi pengembangan di sisi klien. (5) Sistem berhasil diuji melalui tiga pengujian yaitu pengujian pada fitur *monitoring*, pengujian pada fitur *checkzone*, dan pengujian keseluruhan sistem menggunakan *User Acceptance Test* (UAT).

Adapun saran yang didapatkan dari pengembangan sistem UAVITS Apps berbasis Android bahwa pada fitur *monitoring* menggunakan simbol *drone* yang tepat, agar tidak perlu mengatur arah hadap simbol sesuai dengan pergerakan PUTA.

## DAFTAR PUSTAKA

- [1] R. Amin, M. F. Al Aziz, I. Manalu, and Ishanti Nurjanah Rahayu R, "Aspek hukum pengoperasian pesawat udara tanpa awak menurut hukum positif di Indonesia," *Krtha Bhayangkara*, vol. 16, no. 1, pp. 1–22, 2022, doi: 10.31599/krtha.v16i1.976.
- [2] N. L. Johanssen, S. Alkubti Almasri, and P. A. Hoehner, "Geometry-based UAV MIMO channel modeling and pattern optimization for multimode antennas," *IEEE Trans. Antennas Propag.*, vol. 70, no. 11, pp. 11024–11032, 2022, doi: 10.1109/TAP.2022.3191418.
- [3] Beny, J. Budiman, and A. Nugroho, "Implementasi Geofencing pada Aplikasi Layanan Pemantau Anak Berbasis Lokasi," in *Prosiding 2nd Seminar Nasional IPTEK Terapan (SENIT) 2017*, 2017, vol. 2, no. 1, pp. 63–66.
- [4] I. Kurniawan, Humaira, and F. Rozi, "Rest api menggunakan nodejs pada aplikasi transaksi jasa elektronik berbasis android," *J. Ilm. Teknol. Sist. Inf.*, vol. 1, no. 4, pp. 127–132, 2020, doi: 10.30630/jitsi.1.4.18.
- [5] H. A. Purnama, "Pengembangan dan Maintenance Aplikasi Kesehatan pada PT. Global Urban Esensial," Departemen Informatika, Universitas Atma Jaya Yogyakarta, 2020.
- [6] R. Renaldi, B. Cahyo Santoso, Y. Natasya, S. Willian, and F. Alfando, "Tinjauan pustaka sistematis terhadap basis data MongoDB," *J. Inov. Inform.*, vol. 5, no. 2, pp. 132–142, 2020, doi: 10.51170/jii.v5i2.79.
- [7] Y. Firmansyah, R. Maulana, and M. S. Maulana, "Implementasi metode SDLC prototype pada sistem informasi indeks kepuasan masyarakat (IKM) berbasis website studi kasus dinas kependudukan dan catatan sipil," *J. Sist. dan Teknol. Inf.*, vol. 9, no. 3, pp. 315–323, 2021, doi: 10.26418/justin.v9i3.46964.
- [8] S. Sunandar and H. Herlawati, "Prototype mesin presensi berbasis e-mail pada smp assyairiyah atthahiriyah Jakarta," *J. Students' Res. Comput. Sci.*, vol. 1, no. 2, pp. 179–190, 2020, doi: 10.31599/jsrsc.v1i2.444.