

# Aero-Track: Perangkat Lunak Perekam Data Penerbangan Aeronautika

Muhammad Rifqi Fathan<sup>1</sup>, Aditya<sup>1\*</sup>, Indra Gifari Afriansyah<sup>1</sup>, Rani Silvani Yousnaidi<sup>1</sup>, Rossi Passarella<sup>2</sup>, Osvari Arsalan<sup>1</sup>, Rizki Kurniati<sup>1</sup>, Marsella Vindriani<sup>2</sup>

<sup>1</sup> Jurusan Teknik Informatika, Fakultas Ilmu Komputer, Universitas Sriwijaya

<sup>2</sup> Jurusan Sistem Komputer, Fakultas Ilmu Komputer, Universitas Sriwijaya

Email: aditya728391@gmail.com

**Abstract**— Kecelakaan pesawat terbang bisa terjadi pada semua fase penerbangan. Pada tahun 2014, di Indonesia sendiri sudah terjadi kecelakaan penerbangan berjumlah 84 kali. Maka dari itu, kami mengembangkan sebuah perangkat lunak bernama Aero-Track untuk merekam data penerbangan dengan kriteria spesifik mengenai area dan fase penerbangan. Perangkat lunak ini sudah diuji coba dengan merekam data penerbangan pada bandara Sultan Syarif Kasim II, bandara Sultan Mahmud Badaruddin II dan Bandara Sultan Hasanuddin. Data dari hasil perekaman tersebut sudah dapat dijadikan bahan analisis terkait pola dan karakteristik penerbangan.

**Keywords**— data penerbangan, aero track, take off, landing

## I. PENDAHULUAN

Kecelakaan pesawat terbang bisa terjadi pada semua fase penerbangan. Akan tetapi, kecelakaan pesawat dengan kategori fatal *accidents* paling banyak terjadi saat fase final *approach* sebanyak 28% dan fase *landing* sebanyak 26%. Sedangkan kecelakaan dengan kategori *onboard fatalities* paling banyak terjadi pada fase *cruise* sebanyak 29% dan fase final *approach* sebanyak 34% [1].

Kecelakaan pesawat terbang yang terjadi di Indonesia sendiri telah dibahas oleh penelitian yang berjudul Identifikasi Tingkat Kerawanan Bandar Udara Di Indonesia yang ditulis oleh Sandhavitri dkk pada tahun 2014. Jumlah kecelakaan penerbangan yang terjadi di Indonesia berjumlah 84 kecelakaan dan jumlah pergerakan dari penerbangan berjumlah 6.714.893 terhitung dari tahun 2006 hingga tahun 2011[2]. Penelitian ini juga memaparkan 10 bandara paling berbahaya di Indonesia berdasarkan nilai bahaya yang didapat melalui nilai deviasi masing-masing bandara.

Berdasarkan beberapa data yang disebutkan di atas, potensi terjadinya kecelakaan pesawat terbang masih sangat mungkin sering terjadi, maka kami membuat sebuah perangkat lunak untuk merekam data-data pesawat terbang yang memungkinkan untuk mengumpulkan data dengan kriteria spesifik terkait bandara tertentu dan fase yang sedang dialami oleh pesawat yaitu *landing* atau *take off* agar dapat menjadi sumber data bagi penelitian yang dapat menjadi referensi untuk menjadi bahan evaluasi dalam meningkatkan kualitas dan keamanan transportasi pesawat terbang.

## II. METODE

### A. Metode Pengambilan Data

Dalam proses pengambilan data. Peneliti menggunakan sebuah aplikasi yang memiliki nama Aero-

*Track*. Dari data pesawat terbang yang terekam melalui teknologi ADS-B [3], [4], data pesawat terbang tersebut disimpan oleh *receiver* milik Flightradar24 yang kemudian, data tersebut diambil menggunakan aplikasi Aero-Track melalui API yang disediakan oleh Flightradar24 [5]. API yang disediakan oleh Flightradar24 memungkinkan pengguna untuk mendapatkan data pesawat terbang dengan batasan area tertentu. Penentuan batasan area tersebut dapat dilakukan dengan memasukkan 4 buah garis berupa *latitude* dan *longitude* bumi pada *link API*. Berikut contoh pembatasan area pada API Flightradar24 pada Gambar 1.



Gambar 1. API Flightradar24

Dari aplikasi Aero-Track tersebut, data-data yang sudah didapatkan melalui API Flightradar24 akan disimpan pada *database* dan ditampilkan dalam bentuk pergerakan pesawat yang sedang direkam di atas peta.

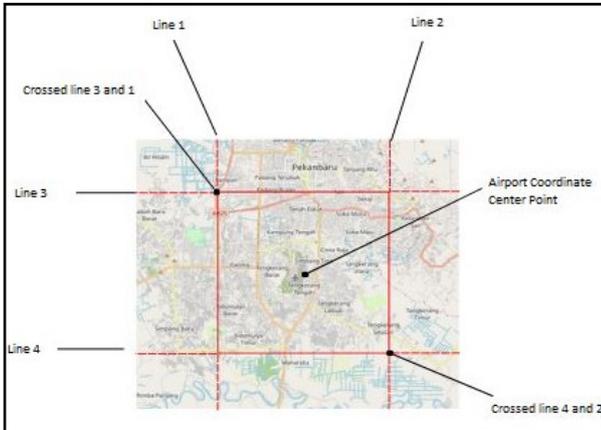
### B. Metode Pengembangan Perangkat Lunak

Pengembangan perangkat lunak pada penelitian ini akan menerapkan metode pengembangan *Rational Unified Process* (RUP). Adapun fase-fase yang akan dilakukan dalam penerapan yaitu fase *inception*, *elaboration*, *construction*, dan *transition* [6]. Fase-fase tersebut akan dijelaskan sebagai berikut :

- Fase *inception*. Pada tahap awal ini, peneliti menetapkan kebutuhan-kebutuhan serta fitur-fitur yang akan diperlukan pada perangkat lunak. Kebutuhan-kebutuhan divisualisasikan dengan bentuk rancangan *usecase*.
- Fase *elaboration*. Pada tahap ini, peneliti menentukan arsitektur perangkat lunak serta desain *interface* sesuai dengan *usecase* yang sudah dibuat pada fase sebelumnya. Jika membutuhkan penambahan fitur, maka peneliti melakukan penyesuaian *usecase*. Kami menggunakan bahasa pemrograman PHP dengan penerapan *CodeIgniter* dalam mengembangkan perangkat lunak.
- Fase *construction*. Pada fase *construction* peneliti melakukan konstruksi perangkat lunak dan membuat kasus uji untuk setiap *usecase* dan melakukan perbaikan jika terdapat kesalahan dalam pengujian.
- Fase *transition*. Tahap terakhir peneliti melakukan testing untuk melihat performa dari perangkat lunak.

### C. Pembatasan Wilayah Perekaman (Boundary)

Pada proses pembuatan *boundary*, perlu dilakukan perhitungan untuk menentukan luasan area yang akan dijadikan target perekaman data pesawat, untuk contoh di bawah ini, kami membuat *boundary* yang hanya mencakup seluruh landasan pacu pada bandara. Ada beberapa komponen yang menjadi patokan bagi peneliti untuk menentukan *boundary* pada bandara yaitu, garis 1, garis 2, garis 3, garis 4, titik perpotongan garis 3 dan garis 1, titik perpotongan garis 4 dan garis 2, titik pusat bandara. Dapat dilihat contoh dari visualisasi komponen pada bandara Sultan Syarif Kasim II [7] pada Gambar 2.



Gambar 2. Komponen *Boundary* pada Aero-Track

Pembatasan area tertentu dapat dilakukan dengan mengambil 4 garis berdasarkan garis lintang dan garis bujur bumi sehingga membentuk bujur sangkar atau persegi panjang dari 4 titik perpotongan garis – garis tersebut yang dapat dilihat pada gambar. Penentuan garis-garis tersebut memanfaatkan 2 titik perpotongan garis yang membentuk bujur sangkar yaitu titik perpotongan garis 3 dan 1 dan titik perpotongan garis 4 dan 2. Untuk mendapatkan garis 1 dan garis 2 Peneliti menggunakan bantuan dari aplikasi *Google Earth*.

Kemudian untuk menentukan koordinat *longitude* *latitude* dari titik perpotongan garis 4 dan 2 peneliti mencari titik refleksi dari titik perpotongan garis 3 dan 1 tersebut terhadap titik pusat bandara. Perpotongan garis atas kiri kita sebut sebagai titik A, titik pusat bandara kita sebut sebagai titik P dan refleksi dari titik A terhadap titik P adalah titik A'[8].

Diketahui :

$$\text{Latitude A} = x = 0.510786$$

$$\text{Longitude A} = y = 101.395$$

$$\text{Latitude P} = a = 0.460786$$

$$\text{Longitude P} = b = 101.445$$

Maka :

$$\text{Latitude A}' = x' = 2a - x$$

$$= 2 \times 0.460786 - 0.510786$$

$$= 0.410786$$

$$\text{Longitude A}' = y' = 2b - y$$

$$= 2 \times 101.445 - 101.395$$

$$= 101.495$$

Dari Perhitungan di atas didapatkan koordinat titik perpotongan antara garis 4 dan 2 adalah 0.410786 , 101.495.

Dari proses penentuan *boundary* yang sudah dijelaskan di atas maka didapatkan seluruh garis yang dibutuhkan untuk membentuk area perekaman data pesawat terbang yaitu :

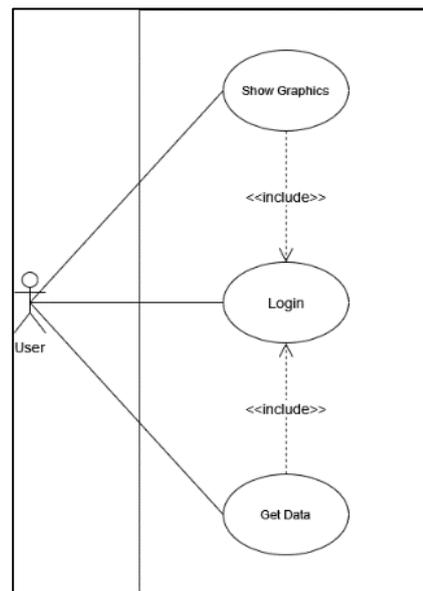
- Garis 1 : 101.395
- Garis 2 : 101.495
- Garis 3 : 0.510786
- Garis 4 : 0.410786

### III. RANCANGAN PENGEMBANGAN PERANGKAT LUNAK

Pada Bagian ini, dipaparkan mengenai rancangan detail perangkat lunak melalui diagram-diagram dan penjelasan detail, sub-bagian yang akan dipaparkan yaitu, *Usecase*, *Scenario Usecase*, Rancangan *Database*, dan *Flowchart* [9]–[11]

#### A. *Usecase* dan *Scenario Usecase*

*Usecase* aplikasi *Aero Track* terdiri dari 3 *usecase* yang dapat dilihat pada Gambar 3. Secara garis besar, *usecase Login* adalah proses saat user melakukan *input username* dan *password* agar dapat masuk ke dalam sistem serta mendapatkan akses penuh untuk seluruh fitur, *usecase Get Data* Melakukan perekaman data pada bandara yang telah dijadikan target perekaman data, dan *Show Graphics* Menampilkan visualisasi data dalam bentuk grafik dengan kriteria tanggal, jenis pesawat, dan jenis maskapai terhadap data yang telah direkam.



Gambar 3. *Usecase* Aero-Track

Penjelasan detail mengenai deskripsi, tujuan hingga langkah-langkah dari masing-masing *usecase* dipaparkan dalam bentuk *scenario usecase* pada Tabel I - III di bawah ini.

TABEL I. SCENARIO USE CASE LOGIN

Identifikasi	
Nomor	01
Nama	Login
Tujuan	Mendapatkan akses penuh untuk melihat atau mengelola data
Deskripsi	Melakukan input username dan password untuk dapat masuk kedalam sistem
Aktor	User
Skenario Utama	
Kondisi Awal	Halaman login terbuka
Aksi Aktor	Reaksi Sistem
1. Menekan tombol Sign In	
	2. Sistem melakukan pengecekan username di database
	3. Sistem melakukan validasi password dengan username di database
Skenario Alternatif A	
Aksi Aktor	Reaksi Sistem
1. Menekan tombol Sign In	
	2. Sistem tidak menemukan username di database
	3. Menampilkan pesan username tidak ditemukan
Skenario Alternatif B	
Aksi Aktor	Reaksi Sistem
1. Menekan tombol Sign In	
	2. Sistem melakukan pengecekan username di database
	3. Sistem tidak menemukan kecocokan username dan password
	4. Menampilkan pesan password tidak valid
<b>Kondisi Akhir</b>	User masuk ke halaman home

TABEL II SCENARIO USE CASE GET DATA

Identifikasi	
Nomor	02
Nama	Get Data
Tujuan	Merekam data pesawat terbang
Deskripsi	Melakukan proses perekaman data dan menyimpan data di dalam database.
Aktor	User
Skenario Utama	
Kondisi Awal	Data sudah difilter
Aksi Aktor	Reaksi Sistem
Memilih bandara target	
	Mengakses database dan membuat Batasan area perekaman
	Menyimpan data perekaman ke database
	Menampilkan data berupa pesawat diatas peta.
<b>Kondisi Akhir</b>	Data tersimpan pada database

TABEL III. SCENARIO USE CASE SHOW GRAPHIC

Identifikasi	
Nomor	03
Nama	Show Graphic
Tujuan	Menampilkan grafik transisi altitude data
Deskripsi	Menampilkan visualisasi transisi altitude data dalam bentuk grafik garis.
Aktor	User
Skenario Utama	
Kondisi Awal	Halaman Home terbuka
Aksi Aktor	Reaksi Sistem
1. Menentukan Range waktu	
2. Menentukan jenis Aircraft	
3. Menentukan jenis Maskapai	
	4. Menampilkam Grafik
<b>Kondisi Akhir</b>	Grafik tertampil

### B. Rancangan Database

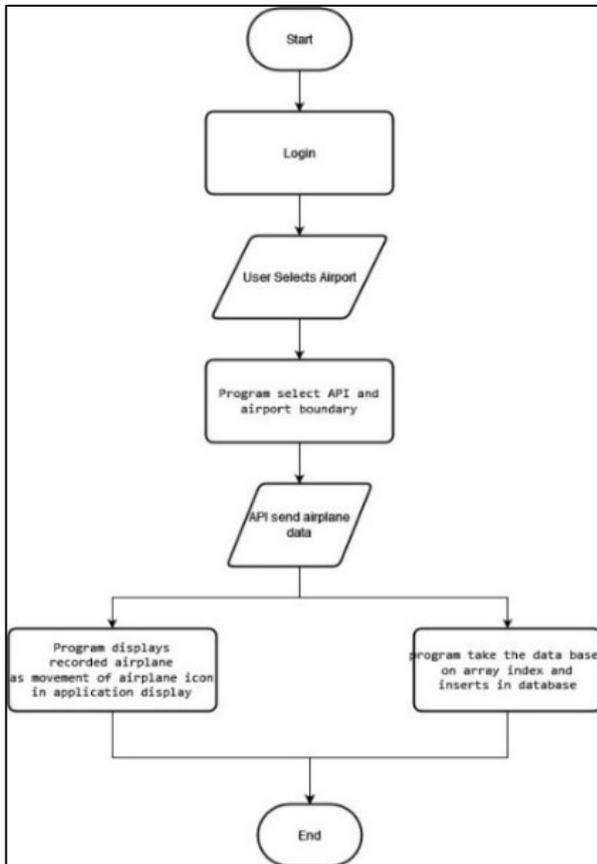
Struktur *database* terdiri dari 3 tabel yang itu tabel *user*, tabel *airport* dan tabel *flight*. Tabel *user* berfungsi sebagai penyimpanan data akun pengguna yang akan digunakan untuk memvalidasi proses otentikasi data yang dimasukkan pengguna saat melakukan proses *login*. Tabel *user* sendiri terdiri dari 4 atribut yaitu *id*, *nama*, *email*, dan *password*.

Tabel selanjutnya adalah tabel *airport*. Tabel ini berfungsi untuk menyimpan data bandara. Bandara – bandara yang tersimpan pada tabel ini, nantinya dapat dipilih oleh pengguna untuk menentukan target perekaman data pesawat. Tabel *airport* memiliki 11 atribut yang memuat informasi identitas, koordinat maupun link API *flightradar24* pada bandara. Informasi yang dimuat pada atribut-atribut tabel *airport* juga berfungsi untuk menentukan luas *boundary* dari bandara. Akan tetapi, seluruh bandara yang termuat pada tabel ini, masih memiliki luas *boundary default*. yaitu *id*, *code\_airport*, *lat1*, *lng1*, *lat2*, *lng2*, *name*, *center\_lat*, *center\_lng*, dan *link*.

Tabel yang terakhir adalah tabel *flight*. Tabel *flight* merupakan tabel untuk menampung data dari pesawat yang direkam melalui index array yang dikirim dari API. Tabel ini memiliki 21 atribut untuk menampung data pesawat yaitu, *flight\_id*, *tanggal*, *icao24*, *latitude*, *longitude*, *heading*, *altitude*, *ground\_speed*, *squawk*, *radar*, *aircraft\_code*, *registration*, *time*, *departure*, *destination*, *number*, *airline\_iata*, *on\_ground*, *vertical\_speed*, *callsign*, dan *airline\_icao*.

### C. Flowchart

Pada bagian ini, akan dijelaskan alur kerja dari aplikasi ini mulai dari *login* hingga data tersimpan dalam *database* dan ditampilkan. Alur secara garis besar dapat dilihat pada Gambar 4.



Gambar 4. Flowchart perekaman data Aero-Track

#### 1) Login

Pada tahap pertama pengguna diharuskan untuk login menggunakan akun yang sudah disediakan oleh admin dan disimpan dalam *database*. Untuk masuk ke aplikasi ini, pengguna diharuskan memasukkan *username* dan *password* pada tampilan *login*.

#### 2) User Select Airport

Saat ingin melakukan perekaman data tentunya pengguna harus memilih bandara yang akan dijadikan target perekaman data. Aplikasi akan menyesuaikan bentuk *boundary* area perekaman data pada bandara yang dipilih.

#### 3) Program Select API and Airport Boundary

Setelah Pengguna memilih bandara yang akan dijadikan target perekaman data pesawat terbang. Aplikasi akan mengambil data *API* yang sudah dilengkapi dengan *boundary* dari bandara tersebut berdasarkan luasan area yang ditentukan oleh admin yang tersimpan di dalam *database*. Pada fitur ini, luas *boundary* yang ada pada bandara, diatur secara *default* dan memiliki ukuran *boundary* yang sama untuk setiap bandara.

#### 4) API Send Airplane Data

Setelah memilih bandara, aplikasi akan memasukkan data *boundary* pada API Flightradar24. Data yang masuk dari API Flightradar24 akan berbentuk *array* yang memuat 19 indeks dari 0-18 yang memuat informasi mengenai pesawat terbang. Berikut contoh dari salah satu *API* pada Gambar 5

full_count:	16242
version:	4
▼ 2db0aee6:	
0:	"8A0243"
1:	-6.2551
2:	106.5658
3:	100
4:	6625
5:	264
6:	""
7:	"F-WIII4"
8:	"B738"
9:	"PK-GFI"
10:	1664731417
11:	"CGK"
12:	"DJJ"
13:	"GA656"
14:	0
15:	2560
16:	"GIA656"
17:	0
18:	"GIA"

Gambar 5. Data API Flightradar24

#### 5) Program Displays Recorded Airplane As Movement Of Airplane Icon In Application Display

Aplikasi ini akan menampilkan pergerakan pesawat yang direkam di atas peta saat pesawat tersebut masuk ke *area boundary*.

#### 6) Program Take the Data Based on Array Index And Inserts In Database

Aplikasi akan mengambil data-data tersebut berdasarkan indeks dan kemudian memasukkan pada tabel yang sudah tersedia yang kemudian akan disimpan dalam bentuk *database*. Data yang sudah disimpan ke *database* akan memiliki variabel – variabel yaitu ; *flight\_id*, tanggal, *icao24*, *latitude*, *longitude*, *heading*, *altitude*, *ground\_speed*, *squawk*, *radar*, *aircraft\_code*, *registration*, *time*, *departure*, *destination*, *number*, *airline\_iata*, *on\_ground*, *vertical\_speed*, *callsign*, dan *airline\_icao*.

## IV. HASIL PENGUJIAN PERANGKAT LUNAK

Perangkat lunak ini sudah diuji coba dengan merekam data pesawat dari beberapa bandara. Adapun hasil dari perekaman sebagai berikut;

- Penerbangan yang sedang melakukan proses *landing* pada bandara Sultan Syarif kasim II selama 3 bulan mulai dari 26 September hingga 26 Desember 2022. Data yang terekam sebanyak 96.941 data dari 1331 penerbangan
- Penerbangan yang sedang melakukan proses *take off* pada bandara Sultan Hasanuddin selama 3 bulan mulai dari 26 September hingga 26 Desember 2022. Data yang terekam sebanyak 192,622 data dari 4,711 penerbangan
- Penerbangan yang sedang melakukan proses *take off* pada bandara Sultan Mahmud Badaruddin II selama 3 bulan mulai dari 1 Februari hingga 1 Agustus 2022 Data yang terekam sebanyak 650,564 data dari 3,148 penerbangan

## V. KESIMPULAN

Dari data yang sudah didapatkan pada beberapa bandara seperti yang dijelaskan pada bagian hasil dan pengujian perangkat lunak, data-data tersebut sudah dapat dijadikan sebagai bahan analisis terkait pola dan karakter penerbangan.

### DAFTAR PUSTAKA

- [1] Boeing, "Statistical Summary of Commercial Jet Airplane Accidents," 2021.
- [2] A. Sandhyavitri, T. Tjahjono, And A. R. Khairumusa, "Identifikasi Tingkat Kerawanan Bandar Udara Di Indonesia," Vol. 14, No. 1, Pp. 43–52, 2014.
- [3] Y. Nurhayati and Susanti, "Implementasi Automatic Dependent Surveillance Broadcast di Indonesia," *Warta Ardhi* *Jurnal Perhubungan Udara*, Vol. 40, No. 3, Pp. 147–162, 2014.
- [4] Icao, "Ads-B Implementation and Operations Guidance Document International Civil Aviation Organization Asia and Pacific Office Cns Sg/22 Appendix K To the Report," 2018.
- [5] Flightradar24, "About Flightradar24," 2023.
- [6] S. Endang Anjarwani, "Penerapan Metode Rational Unified Process (RUP) Dalam Pengembangan Sistem Informasi Medical Check Up Pada Citra Medical Centre," 2020. [Online]. Available: <http://jtika.if.unram.ac.id/index.php/jtika/>
- [7] Jeppesen, "Airport Information For WIBB" Jun. 2022.
- [8] Yohanes Dian Tri Nugroho, "Proses Dan Hasil Belajar Siswa pada Pengembangan pembelajaran Matematikayang Memotivasi keterlibatan Aktifsiswakelas Xii Ipa 4sma Kolese De Brittoyogyakarta Tahun Ajaran 2011/ 2012," 2012.
- [9] S. Hanum, "Pemanfaatan Aplikasi Penggambar Diagram Alir (Flowchart) Sebagai Bahan Ajar Untuk Mata Kuliah Sistem Akuntansi Di Fakultas Ekonomi Pada Perguruan Tinggi Swasta Di Kota Medan," 2017.
- [10] A. Fantechi, S. Gnesi, G. Lami, And A. Maccari, "Applications of Linguistic Techniques For Use Case Analysis," *Requir Eng*, Vol. 8, No. 3, Pp. 161–170, 2003, Doi: 10.1007/S00766-003-0174-0.
- [11] Finandhita Alif, "Pengenalan UML Dan Diagram Use Case," Diakses dari <https://repository.unikom.ac.id/51350/1/Bab%20%20-%20Pengenalan%20UML%20dan%20Diagram%20Use%20Case.pdf>