BAB I

PENDAHULUAN

A. Latar belakang

Seiring dengan perkembangan teknologi, transportasi yang ada di Indonesia juga semakin berkembang membuat masyarakat lebih memilih transportasi yang cepat. Sarana transpotasi tidak terlepas dari adanya kemajuan teknologi dari hasil pemikiran manusia sebagai upaya memberikan pelayanan transportasi yang terbaik, aman dan nyaman. Salah satunya adalah transportasi udara, transportasi udara ini lah yang menjadi prioritas masyarakat saat ini untuk berpergian maupun untuk pengiriman barang, (Sherina Kumala Dewi. 2023)

Saat ini banyak penumpang yang melakukan perjalanan antar kota, antar daerah bahkan antar Negara dengan menggunakan pesawat udara. Setiap penumpang ketika bepergian tentunya membawa barang-barang dimana barang-barang ini akan dimasukkan dalam sebuah bagasi. Bagasi yang dibawa oleh penumpang akan dimasukkan ke dalam pesawat udara dimana bagasi tersebut telah sesuai catatan standar bagasi, seperti bagasi yang beratnya sesuai dengan prosedur, bagasi telah terkunci dan bagasi yang dibungkus rapi untuk menghindari kerusakan ataupun kehilangan bagasi, (Inneke Febryana. 2022).

Dalam hal ini setiap perusahaan tentunya tidak menginginkan terjadinya sebuah hal yang bisa berakibat fatal.

Di Papua khususnya Kabupaten Yahukimo transportasi terdiri dari

transportasi darat, transportasi udara, transportasi laut. Transportasi udara memiliki keunggulan dibanding transportasi lainnya. Karena transportasi yang satu ini lebih cepat, sehingga menjadikan transportasi udara sebagai pilihan pertama untuk melakukan perjalanan jauh, (Badan Pusat Statistik Kabupaten Yahukimo).

Dari hasil penelusuran, Bandara NOP Goliat Dekai merupakan salah satu bandara tersibuk di Indonesia dengan jumlah penumpang yang terus meningkat setiap tahunnya. Seiring dengan meningkatnya jumlah penumpang, permasalahan terkait kehilangan bagasi juga semakin sering terjadi. Kehilangan bagasi tidak hanya menyebabkan ketidaknyamanan bagi penumpang, tetapi juga dapat berpotensi menimbulkan kerugian finansial dan emosional.

Saat ini, proses pelaporan kehilangan bagasi di Bandara NOP Goliat Dekai masih dilakukan secara manual dengan mengisi formulir kertas dan menunggu proses verifikasi yang cukup lama, sistem ini dinilai kurang efesien dan rentan terhadap kesalahan manusia, seperti hilangnya data, kesalahan penulisan, atau bahkan kehilangan formulir itu sendiri. Selain itu, proses manual juga menyulitkan dalam melacak dan memantau status laporan, serta memberikan update terbaru kepada penumpang yang kehilangan bagasi.

Dengan demikian, aplikasi ini akan memungkinkan penumpang untuk melaporkan kehilangan bagasi secara online, baik melalui komputer maupun perangkat seluler. Aplikasi ini akan menyediakan antarmuka yang mudah digunakan, sehingga peumpang dapat dengan cepat dan akurat melaporkan detail kehilangan bagasi mereka dan memantau status laporan kehilangan bagasi secara

real-text.

Dengan adanya **Aplikasi Penanganan Pelaporan Kehilangan Bagasi di Bandara NOP Goliat Dekai Berbais Web**, diharapkan dapat meningkatkan Kepuasan penumpang, memberikan pelayanan yang lebih baik, serta meningkatkan efesiensi dan produktivitas dalam penanganan kasus kehilangan bagasi di Bandara NOP Goliat Dekai.

B. Rumusan Masalah

1. Bagaimana merancang aplikasi berbasis web yang efektif untuk melaporkan dan menangani kehilangan bagasi di bandara?
2. Bagaimana meningkatkan efisiensi operasional dalam proses pelaporan dan penanganan kehilangan bagasi melalui aplikasi berbasis web?
3. Bagaimana aplikasi dapat memberikan informasi *real-time* dan transparan kepada penumpang mengenai status pencarian bagasi yang hilang?

C. Tujuan Penelitian

Tujuan yang ingin dicapai dalam penelitian ini, yakni membuat aplikasi berbasis web untuk Penanganan Pelaporan Kehilangan Bagasi di bandara Nop Goliat Dekai agar dapat mempermudah penumpang melaporkan kehilangan bagasi secara cepat dan mudah, juga meningkatkan kepuasan dan kepercayaan penumpang terhadap layanan bandara.

D. Batasan Masalah

Untuk memastikan penelitian ini tetap fokus dan dapat diselesaikan dalam waktu yang ditentukan, beberapa batasan masalah ditetapkan sebagai berikut:

1. Ruang Lingkup Aplikasi: Penelitian ini dibatasi pada Aplikasi Berbasis Web Untuk Penanganan Pelaporan Kehilangan Bagasi Di Bandara Nop Goliat Dekai. Aplikasi ini hanya akan mencakup fitur-fitur yang berkaitan langsung dengan proses pelaporan, pelacakan, dan penanganan kasus kehilangan bagasi. Fitur-fitur lain yang tidak berhubungan langsung dengan penanganan kehilangan bagasi, seperti informasi penerbangan atau layanan bandara lainnya, tidak termasuk dalam cakupan penelitian ini.
2. Pengguna dan Akses: Aplikasi dirancang untuk digunakan oleh dua kelompok pengguna utama: staf bandara yang menangani laporan kehilangan bagasi dan penumpang yang melaporkan kehilangan bagasi. Akses ke aplikasi akan dibatasi hanya untuk pengguna yang terdaftar, dengan hak akses yang berbeda berdasarkan peran pengguna. Penelitian ini tidak mencakup integrasi dengan sistem manajemen identitas bandara yang lebih luas atau sistem autentikasi pihak ketiga.

Batasan-batasan ini ditetapkan untuk memastikan penelitian dapat dilakukan secara efektif dan efisien, sesuai dengan sumber daya dan waktu yang tersedia. Meskipun demikian, hasil penelitian ini diharapkan dapat memberikan dasar yang kuat untuk pengembangan dan implementasi lebih lanjut dimasa depan.

E. Manfaat Penelitian

Adapun manfaat yang dapat diberikan dari penelitian ini adalah sebagai berikut:

1. Manfaat Akademis
    a. Penelitian ini diharapkan dapat menambah literatur dalam bidang teknologi informasi dan manajemen penerbangan.
    b. Penelitian ini diharapkan dapat memberikan kontribusi pada inovasi dibidang teknologi informasi, khususnya dalam aplikasi yang mendukung pelayanan publik.
2. Manfaat Bagi Penumpang
    a. Penumpang dapat melaporkan kehilangan bagasi secara cepat dan mudah melalui aplikasi berbasis web, tanpa perlu mengantre atau mengisi formulir manual.
    b. Penumpang dapat menerima update secara *real-text* tentang status pelaporan dan pencarian bagasi yang hilang.
3. Manfaat Bagi Maskapai Penerbangan
    a. Aplikasi ini dapat membantu maskapai mengelola dan memproses laporan kehilangan bagasi dengan lebih efisien, mengurangi waktu dan sumber daya yang dibutuhkan dibandingkan dengan proses manual.
    b. Dengan sistem yang lebih cepat dan transparan, maskapai dapat meningkatkan tingkat kepuasan pelanggan, yang berdampak positif pada reputasi mereka.

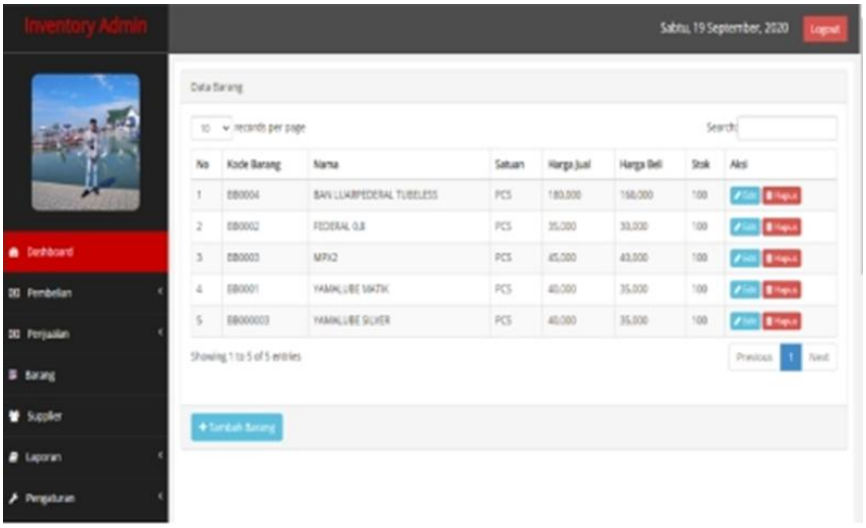
BAB II

TINJAUAN PUSTAKA

A. Kajian teori

1. Aplikasi

Aplikasi merupakan suatu *sub* kelas perangkat lunak komputer yang memanfaatkan kemampuan komputer langsung untuk melakukan suatu tugas yang diinginkan pengguna.



Gambar 2.1 Aplikasi

2. Aplikasi berbasis Web

Aplikasi web merupakan sebuah aplikasi yang mengunakan teknologi browser untuk menjalankan aplikasi dan diakses melalui jaringan komputer. Sedangkan menurut (Limbong, t. 2021), aplikasi web adalah sebuah program yang

tersimpan dalam *server* dan dikirim melalui internet serta diakses melalui antarmuka *browser*. Dari pengertian di atas, dapat disimpulkan aplikasi web merupakan aplikasi yang diakses mengunakan *web browser* melalui jaringan internet atau intranet. Aplikasi web juga merupakan suatu perangkat lunak komputer yang dikodekan dalam bahasa pemrograman yang mendukung perangkat lunak berbasis web seperti *HTML, JavaScript, CSS, Ruby, Python, PHP, Java* dan bahasa pemrograman lainnya.

Ada dua bagian pokok dalam aplikasi web, pertama adalah sisi *client* dan kedua adalah sisi *server*. Sisi *client* dalam hal ini adalah PC atau bisa juga perangkat *mobile* yang terhubung ke jaringan internet. *Client* dapat mengakses aplikasi web melalui *web browser* seperti internet *explorer, Mozilla firefox, Opera* dan lain-lain, sedangkan *server* adalah perangkat komputer dengan spesifikasi yang bagus digunakan untuk menyimpan aplikasi web beserta *database server* yang siap untuk diakses oleh *client*. *Client* bertugas meminta halaman *web server* melalui *web browser*, kemudian *web browser* akan meneruskan ke *server* dimana aplikasi web berada. *Computer server* akan mengolah permintaan dari *client*, ketika halaman web yang diminta ditemukan maka *computer server* akan mengirimkannya ke *computer client* dan halaman web yang diminta akan ditampilkan pada *web browser* di *computer client*.

Beberapa keunggulan yang dimiliki aplikasi berbasis web, antara lain:

(1) Aplikasi berbasis web dapat dijalankan dimanapun kapanpun tanpa harus melakukan penginstalan.

(2) Terkait dengan isu lisensi (hak cipta), kita tidak memerlukan lisensi ketika menggunakan *web-based application*, sebab lisensi telah menjadi tanggung jawab dari web penyedia aplikasi.

(3) Dapat dijalankan disistem operasi manapun seperti menggunakan *linux* maupun *windows*, aplikasi berbasis web dapat dijalankan asalkan kita memiliki browser dan akses internet.

(4) Dapat diakses melalui banyak media seperti komputer, *handheld* dan *handphone* yang sudah sesuai dengan standar WAP.

(5) Tidak perlu spesifikasi komputer yang tinggi untuk menggunakan aplikasi berbasis web ini, sebab dibeberapa kasus, sebagian besar proses dilakukan di web server penyedia aplikasi berbasis web ini.

Selain memiliki keunggulan, terdapat beberapa kekurangan yang perlu diperhatikan pada aplikasi berbasis web, antara lain:

a. Dibutuhkan koneksi intranet dan internet yang handal dan stabil. Hal ini bertujuan agar aplikasi dapat dijalankan dengan baik dan lancar.

b. Dibutuhkan sistem keamanan yang baik dikarenakan aplikasi dijalankan secara terpusat, sehingga apabila server di pusat mengalami *down*, maka sistem aplikasi tidak bisa berjalan.

3. HTML



Gambar 2. 2 *Hypertext Markup Language*

HyperText Markup Language atau HTML, adalah bahasa markup standar industri untuk membangun halaman web. Ini menawarkan kumpulan tag dan properti yang menentukan pengorganisasian halaman web, konten, dan format halaman web. Untuk merepresentasikan elemen pada halaman web, HTML menggunakan struktur hierarkis yang dikenal sebagai Document *Object Model* (DOM). Teks, foto, tautan, mul*text*dia, dan elemen lainnya semuanya dapat dimuat dalam setiap elemen, yang dibatasi oleh tag pembuka dan penutup.

Istilah "*HyperText*" mengacu pada kemampuan HTML untuk membuat tautan yang memungkinkan pengguna dengan mudah dan cepat berpindah dari satu situs web ke situs web lainnya. *HyperText Markup Language*, juga dikenal sebagai HTML, adalah bahasa global *World Wide Web*. Seperti yang tersirat dari namanya *HyperText Markup Language*, HTML adalah bahasa markup bukan bahasa pemrograman (Oktaviani, N., Widiarta, I. M., & Nurlaily. 2019).

Dokumen yang dibuat dengan menggunakan bahasa bahasa

markup seperti HTML memiliki bagian yang dirender pada hasil akhir serta bagian yang menginstruksikan program rendering cara menginterpretasikan teks yang tersisa. Proses mengubah dokumen teks yang berisi teks HTML menjadi, misalnya, representasi visualnya di layar disebut sebagai "*rendering*" (Sadiku, M. N., Ashaolu, T. J., & Musa, S. M. (2019).

Perkembangan HTML sangat pesat. Dari permulaan HTML 1.0 yang sederhana, yang mencakup elemen dasar seperti judul dan paragraf, hingga penambahan *tabel* dan gambar pada HTML 2.0, dan akhirnya ke frame dan dukungan form yang disempurnakan pada HTML 3.2. Dengan CSS, konten dan presentasi sekarang dapat dipisahkan dalam HTML 4.01, yang juga meningkatkan komponennya. Dengan elemen semantiknya, dukungan mul*text*dia, bentuk yang lebih baik, fitur aksesibilitas, dan kemampuan *mobile-friendly,* HTML5 menandai periode transformasi.

Versi HTML terbaru, yang dikenal sebagai HTML5, digunakan untuk membuat dan mengatur dokumen di internet. Ini mencakup elemen dan fitur baru yang memungkinkan pembuatan halaman web yang lebih dinamis dan menarik, termasuk dukungan untuk audio dan video, kanvas untuk grafik dan animasi, dan peningkatan kompatibilitas dengan perangkat seluler.

Fungsionalitas baru yang diperkenalkan oleh HTML5 memungkinkan pembuatan halaman web yang lebih dinamis dan

interaktif, yaitu:

a. Dukungan audio dan video.
b. Kanvas visual dan animasi.
c. Dukungan yang ditingkatkan untuk *mobile*.
d. *Semantic Element* baru termasuk *nav, article, sections* dan *headers.*
e. Kontrol untuk form, yaitu tanggal, waktu, *email*, pencarian, dll.
f. Penyimpanan web untuk menyimpan informasi secara lokal.
g. Fitur *drag and drop*.

4. CSS



Gambar 2. 3 *cascading style sheets*

CSS adalah bahasa yang terus berkembang untuk menggambarkan presentasi konten web di layar, printer, sintesis ucapan, pembaca layar, dan jendela obrolan. CSS digunakan oleh semua *browser* pada semua ukuran layar disemua jenis perangkat IoT, termasuk ponsel, komputer, video game, televisi, jam tangan, dan konsol otomatis. CSS adalah bahasa pemrograman web yang digunakan untuk mengendalikan tampilan halaman web. CSS memiliki berbagai properti yang dapat digunakan untuk mengatur tampilan halaman web. CSS

terdiri dari dua bagian utama, yaitu deklarasi dan selektor. Deklarasi adalah bagian yang menentukan properti dari suatu halaman web. Selektor adalah bagian yang menentukan elemen web mana yang akan diatur oleh deklarasi. CSS juga memiliki berbagai properti yang dapat digunakan untuk mengatur tampilan halaman web. Beberapa properti CSS yang umum digunakan antara lain:

a. Ukuran: properti ini digunakan untuk mengatur ukuran halaman web, seperti ukuran font, ukuran gambar, dan ukuran jarak antar elemen.
b. Warna: properti ini digunakan untuk mengatur warna halaman web, seperti warna font, warna latar belakang, dan warna border.
c. Font: properti ini digunakan untuk mengatur font elemen web, seperti jenis font, ukuran font, dan gaya font.
d. Tata letak: properti ini digunakan untuk mengatur tata letak elemen web, seperti posisi elemen, ukuran elemen, dan jarak antar elemen.

5. PHP



Gambar 2. 4 *Hypertext Preprocessor*

PHP adalah bahasa *server-side* yang secara khusus digunakan untuk aplikasi web. Bahasa PHP berjalan di sisi *server*, artinya ia berjalan pada *server* dan bisa dimasukkan diantara bahasa HTML (*HyperText Markup Language*). Oleh karena itu, kode PHP tidak akan terlihat lagi dan hasil akhir dalam bentuk HTML akan dikirimkan ke *browser* (Krisbiantoro, d., kom, m., abda'u, p. D., &

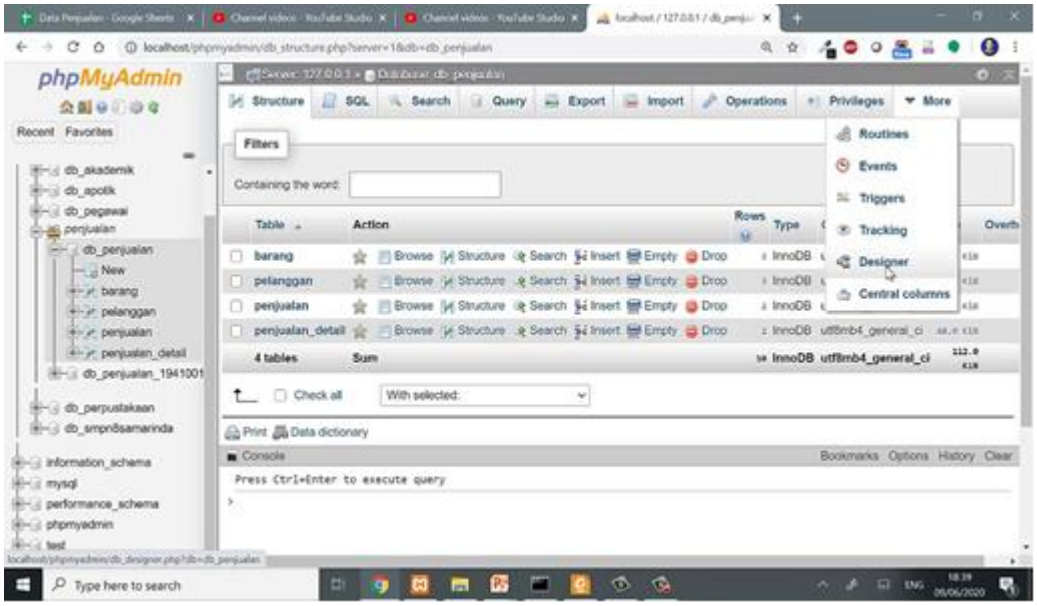
kom, m. 2021). Karena PHP dijalankan di komputer *server*, itu dikenal sebagai bahasa pemrograman sisi *server*. Ini berbeda dengan bahasa pemrograman sisi klien, seperti *Javascript*, yang beroperasi pada klien, yaitu web browser.

Pada awalnya, PHP adalah singkatan dari *Personal Home Page*. PHP digunakan untuk mengembangkan situs web pribadi, sesuai dengan namanya. Seiring berjalannya waktu, PHP berkembang menjadi bahasa pemrograman web yang sangat kuat, tidak hanya digunakan untuk membuat halaman web dasar juga untuk situs-situs populer seperti Wikipedia, WordPress, Joomla, dan lainnya yang dikunjungi oleh jutaan orang.

6. **MySQL**

MySQL adalah *software RDBMS* (atau *server database*) yang dapat mengelola *database* dengan sangat cepat, dapat menampung data dalam jumlah sangat besar, dapat diakses oleh banyak *User* dan dapat melakukan suatu proses secara sinkron atau berbarengan (Limbong, t, 2021).

Sebuah website yang dinamis membutuhkan tempat penyimpanan data agar pengunjung dapat memberi komentar, saran, dan masukan atas website yang dibuat (Fitri, r., kom, s., & kom, m. 2020). Tempat penyimpanan data berupa informasi dalam sebuah *tabel* disebut dengan *database*. Program yang digunakan untuk mengolah dan mengelola *database* adalah MySQL yang memiliki kumpulan prosedur dan struktur sedemikian rupa sehingga memper mudah dalam menyimpan, mengatur, dan menampilkan data. MySQL adalah salah satu *DataBase Management System* (DBMS) dari sekian banyak DBMS.



Gambar 2.5 MYSQL

7. XAMPP

XAMPP adalah Singkatan dari X (tempat sistem operasi apapun), *Appache, MySQL, PHP* dan *Perl. XAMPP* adalah perangkat lunak bebas yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program. *XAMPP* merupakan alat yang menyediakan paket perangkat lunak ke dalam satu buah paket..

8. phpmyadmin



Gambar 2. 6 *phpmyadmin*

Phpmyadmin adalah sebuah aplikasi open source yang berfungsi

untuk memudahkan manajemen *MySQL*. Aplikasi ini memungkinkan pengguna mengelola database MySQL melalui antarmuka berbasis web, sehingga operasi seperti membuat *database*, *tabel*, memasukkan data, menghapus, dan memperbarui data dapat dilakukan dengan lebih mudah tanpa perlu menuliskan perintah SQL secara manual (Wijianto et al., 2018).Dengan menggunakan *Phpmyadmin*, anda dapat membuat *database*, membuat *tabel*, meng-insert, menghapus dan meng-update data dengan GUI dan terasa lebih mudah, tanpa perlu mengetikkan perintah SQL secara manual. Karena berbasis web, maka *Phpmyadmin* dapat dijalankan dibanyak OS, selama dapat menjalankan web *server* dan *MySQL*.

9. Unified modelling language

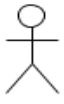
UML adalah serangkaian alat yang digunakan untuk menyederhanakan suatu sistem atau perangkat lunak berbasis objek. Singkatan dari *Unified Modelling Language*, UML juga berperan dalam mempermudah rancang bangun aplikasi secara berkelanjutan. Sistem atau aplikasi yang tidak didokumentasikan dengan baik seringkali menghambat proses rancang bangun karena memaksa para pengembang untuk melakukan penyelidikan dan memahami kode program yang kompleks. UML juga berfungsi sebagai alat untuk men-transfer pengetahuan tentang suatu sistem atau aplikasi dari satu pengembang ke pengembang lainnya. Penggunaan UML dalam tahapan analisis, desain, pengkodean, dan pengujian perangkat lunak membantu pengembang dalam menggambarkan kebutuhan sistem secara efektif (Rahimudin, Widiya afni Aprilia. (2023). Keberadaan UML tidak hanya mempermudah komunikasi

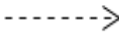
antara para pengembang, tet*API* juga memungkinkan orang di luar lingkaran rancang bangun, seperti pihak bisnis atau siapapun, untuk memahami suatu sistem dengan lebih baik.

UML (Unified Modelling Language), sebagai standar industri, digunakan untuk menggambarkan, merencanakan, dan mendokumentasikan sistem perangkat lunak. UML menyediakan format standar untuk merancang model sistem. Namun, karena UML memanfaatkan kelas dan operasi sebagai bagian dari konsep intinya, lebih sesuai digunakan untuk rancang bangun perangkat lunak dengan bahasa berbasis objek seperti C++, Java, C#, atau VB.NET. Meskipun begitu, UML masih bisa digunakan untuk memodelkan aplikasi prosedural dalam bahasa seperti VB atau C.

UML mulai dikembangkan oleh Object *Management Group* sejak versi 1.0 dirilis pada bulan Januari 1997. Dalam rancang bangun berbasis objek, terdapat beberapa prinsip kunci yang harus dipahami, seperti Objek, Kelas, Abstraksi, Enkapsulasi, Pewarisan, dan Polimorfisme. Adapun daftar simbol UML yaitu:

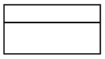
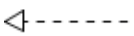
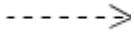
***Tabel* 2. 1** *Symbol Use Case Diagram*

NO.	GAMBAR	NAMA	KETERANGAN
1		*Actor*	Menspesifikasikan himpuan peran yang pengguna mainkan ketika berinteraksi dengan *use case*.
2		*Dependency*	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (*independent*) akan mempengaruhi elemen yang

NO.	GAMBAR	NAMA	KETERANGAN
			bergantung padanya elemen yang tidak mandiri (*independent*).
4		*Generalization*	Menspesifikasikan bahwa *use case* sumber secara eksplisit.
5		*Include*	Menspesifikasikan bahwa *use case* target memperluas perilaku dari *use case* sumber pada suatu titik yang diberikan.
6		*Extend*	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		*Association*	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		*Use Case*	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
9		*Collaboration*	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
10		*Note*	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu

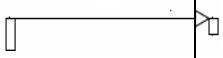
NO.	GAMBAR	NAMA	KETERANGAN
			sumber daya komputasi

***Tabel* 2. 2** *Symbol Class Diagram*

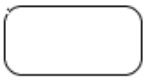
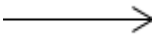
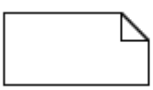
NO.	GAMBAR	NAMA	KETERANGAN
1		*Generalization*	Hubungan dimana objek anak *(descendent)* berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*).
2		*Nary Association*	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		*Class*	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4		*Collaboration*	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu *actor*
5		*Realization*	Operasi yang benar-benar dilakukan oleh suatu objek.
6		*Dependency*	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri *(independent)* akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri.

NO.	GAMBAR	NAMA	KETERANGAN
7		*Association*	Apa yang menghubungkan antara objek satu dengan objek lainnya.

***Tabel* 2. 3** *Symbol Sequence Diagram*

No	GAMBAR	NAMA	KETERANGAN
1		*LifeLine*	Objek *entity*, antarmuka yang saling berinteraksi.
2		*Message*	Spesifikasidari komunikasi antar objek yang memuat in*form*asi-in*form*asi tentang aktivitas yang terjadi
3		*Message*	Spesifikasidari komunikasi antar objek yang memuat in*form*asi-in*form*asi tentang aktivitas yang terjadi.

***Tabel* 2. 4** *Symbol State Chart Diagram*

No	GAMBAR	NAMA	KETERANGAN
1		*State*	Nilai atribut dan nilai *Link* pada suatu waktu tertentu, yang dimiliki oleh suatu objek.
2		*Initial Pseudo State*	Bagaimana objek dibentuk atau diawali.
3		*Final State*	Bagaimana objek dibentuk dan dihancurkan.
4		*Transition*	Sebuah kejadian yang memicu sebuah state objek dengan cara memperbaharui satu atau lebih nilai atributnya..
5		*Association*	Apa yang menghubungkan antara objek satu dengan objek lainnya.
6		*Node*	Elemen fisik yang eksis saat aplikasi dijalankan dan

No	GAMBAR	NAMA	KETERANGAN
			mencerminkan suatu sumber daya komputasi.

***Tabel* 2. 5** *Symbol Activity Diagram*

No.	GAMBAR	NAMA	KETERANGAN
1		*Activity*	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.
2		*Action*	State dari sistem yang mencerminkan eksekusi dari suatu aksi.
3		*Initial Node*	Bagaimana objek dibentuk atau diawali.
4		*Activity Final Node*	Bagaimana objek dibentuk dan dihancurkan..
5		*Fork Node*	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran

B. Kajian hasil penelitian terdahulu

Untuk dapat digunakan sebagai data pendukung, diperlukan berbagai penelitian sebelumnya. Studi sebelumnya yang berkaitan dengan masalah yang dibahas dalam penelitian ini merupakan salah satu informasi pendukung yang peneliti butuhkan. Dalam hal ini, referensi yang berkaitan dengan tantangan teknologi informasi dari studi sebelumnya digunakan. Oleh karena itu penelitian dilakukan secara *online* dengan menggunakan berbagai temuan penelitian yang disajikan dalam makalah akhir, tesis, atau publikasi.

(Faril Aditya Pratama, Febriansyah Ignas Pradana. (2023) dengan judul " Analisis

kinerja *unit lost and found* dalam penanganan bagasi hilang dan rusak maskapai wings air di PT. Merpati angkasa abadi Bandar udara Sultan Muhammad Kaharuddin III Sumbawa". Penelitian ini bertujuan untuk menganalisis kinerja *Unit Lost and Found* di PT Merpati Angkasa Abadi dalam penanganan bagasi hilang dan rusak untuk maskapai Wings Air. Penelitian ini bertujuan untuk mengidentifikasi penyimpangan dari prosedur operasional standar (SOP) dan memberikan rekomendasi untuk meningkatkan kualitas layanan, profesionalisme, dan responsivitas unit dalam menangani masalah penumpang, sehingga dapat meminimalkan ketidakteraturan bagasi dan meningkatkan kepuasan penumpang. Penelitian ini menggunakan pendekatan kualitatif dengan metode pengumpulan data melalui observasi, wawancara, dan dokumentasi. Metode ini dipilih untuk mendapatkan pemahaman yang mendalam tentang proses dan tantangan yang dihadapi oleh unit dalam penanganan bagasi. Hasil penelitian menunjukkan bahwa penanganan bagasi oleh *Unit Lost and Found* belum sepenuhnya sesuai dengan SOP yang ditetapkan. Meskipun petugas berusaha keras, terdapat kekurangan dalam kemampuan yang menyebabkan terjadinya *human error* (Rahmawati, E., & Susanto, H. 2022). Hal ini berkontribusi pada banyaknya kasus penyimpangan bagasi.

(IM Adnyana, Suan Nauli, 2018) dengan Judul "Penggunaan Sistem Aplikasi *Out Of Gauge* Dengan *Baggage Handling System* pada Terminal 3 *Ultimate Domestik* di Bandara Soekarno- Hatta". Penelitian ini berfokus pada perbandingan antara dua sistem penanganan bagasi di Bandara Soekarno-Hatta, yaitu *Baggage Handling System* (BHS) dan *Out Of Gauge* (OOG). Metode yang digunakan dalam penelitian ini adalah data kualitatif yang diperoleh melalui observasi, wawancara, dan studi

pustaka. Tujuan utama dari penelitian ini adalah untuk mengidentifikasi kekuatan dan kelemahan masing-masing sistem penanganan bagasi, serta meningkatkan sinergi antara keduanya untuk efisiensi dan kepuasan pelanggan yang lebih baik. Setelah melakukan penelitian dapat disimpulkan bahwa kedua sistem saling melengkapi dan perlu disinergikan untuk meningkatkan efisiensi dan kepuasan pelanggan. Penelitian ini memberikan wawasan penting tentang bagaimana kedua sistem dapat dioptimalkan untuk meningkatkan pengalaman penanganan bagasi di bandara.

(Djoko Harsono, 2018), Dengan "Perancangan Aplikasi Pelaporan Barang Hilang Di KRL *Commuterline Berbasis Web*". Penelitian ini membahas perancangan aplikasi berbasis web untuk pelaporan barang hilang pada KRL *Commuter Line* di Indonesia, yang dioperasikan oleh PT Kereta Commuter Indonesia. Penelitian ini menyoroti pentingnya teknologi informasi dalam kehidupan modern dan kebutuhan akan sistem yang efisien untuk membantu penumpang dalam melaporkan barang hilang. Dengan menggunakan metode pengembangan berorientasi objek dan *model Rapid Application Development* (RAD), data dikumpulkan melalui wawancara, observasi, dan studi pustaka. Aplikasi ini bertujuan untuk menyederhanakan proses pelaporan, memungkinkan penumpang untuk melaporkan barang hilang secara online dan melacak status laporan mereka, serta memberikan kemudahan bagi staf dalam mengelola barang-barang yang hilang. Penelitian ini juga mengidentifikasi masalah utama dalam sistem manual yang ada dan merumuskan solusi melalui aplikasi baru. Hasil pengujian sistem menunjukkan bahwa aplikasi dapat menangani pendaftaran dengan baik, dan kesimpulan menyatakan bahwa aplikasi ini

mempermudah pelaporan barang hilang, dengan saran untuk meningkatkan keamanan sistem dan mengembangkan aplikasi mobile di masa depan.

C. Kerangka pikir

Penelitian ini berangkat dari identifikasi masalah penanganan kehilangan bagasi di Bandara NOP Goliat Dekai, yang berdampak pada efisiensi operasional dan kepuasan penumpang. Melalui analisis kebutuhan, studi literatur, dan pengumpulan data lapangan, penelitian bertujuan mengembangkan solusi teknologi yang efektif. Fokus utama adalah merancang aplikasi berbasis web yang dapat mengoptimalkan proses pelaporan dan penanganan kasus kehilangan bagasi.

↓

Hasil akhir penelitian adalah aplikasi web yang dirancang khusus untuk penanganan pelaporan kehilangan bagasi di Bandara NOP Goliat Dekai.

↓

Tahap perancangan meliputi tiga komponen utama: desain sistem, desain database, dan desain antarmuka pengguna. Integrasi ketiga aspek ini menghasilkan prototype aplikasi yang kemudian diimplementasikan. Proses pengembangan mengadopsi pendekatan interatif, melibatkan serangkaian pengujian, evaluasi, dan perbaikan untuk memastikan kualitas dan kesesuaian dengan kebutuhan pengguna.

BAB III

METODE PENELITIAN

A. Jenis penelitian

Penelitian ini menggunakan *metode Research and Development* (R&D) atau Penelitian dan Pengembangan. Metode R&D dipilih karena sesuai dengan tujuan utama penelitian ini, yaitu menghasilkan produk berupa aplikasi berbasis web untuk penanganan pelaporan kehilangan bagasi di Bandara NOP Goliat Dekai. Penelitian dan pengembangan adalah proses yang digunakan untuk mengembangkan dan memvalidasi produk pendidikan, yang dalam konteks ini dapat diterapkan pada pengembangan produk teknologi informasi (Oktaviani. N dkk, 2019).

Metode R&D memungkinkan peneliti untuk melakukan serangkaian tahapan sistematis dalam pengembangan aplikasi. Tahapan ini umumnya mencakup analisis kebutuhan, perancangan produk, pengembangan produk, implementasi, dan evaluasi. Setiap tahapan dilakukan secara cermat untuk memastikan bahwa produk akhir yang dihasilkan benar-benar dapat memenuhi kebutuhan pengguna dan mengatasi permasalahan yang ada di lapangan.

Dalam konteks penelitian ini, tahap analisis kebutuhan akan melibatkan pengumpulan data mengenai proses pelaporan kehilangan bagasi yang saat ini berlangsung di Bandara NOP Goliat Dekai. Hal ini dapat dilakukan melalui wawancara dengan staf bandara, observasi langsung, dan analisis dokumen terkait. Hasil dari analisis ini akan menjadi dasar dalam merancang fitur-fitur

yang diperlukan dalam aplikasi web yang akan dikembangkan.

Tahap perancangan akan fokus pada desain antarmuka pengguna (*User* interface) dan arsitektur sistem. Pada tahap ini, peneliti akan merancang diagram alur, diagram entitas-relasi, dan *mockup* aplikasi. Desain ini akan mempertimbangkan aspek *User experience* (UX) untuk memastikan aplikasi mudah digunakan oleh berbagai pengguna, termasuk staf bandara dan penumpang.

Tahap pengembangan melibatkan proses coding atau pemrograman aplikasi berdasarkan desain yang telah dibuat. Dalam hal ini, peneliti akan menggunakan teknologi pengembangan web seperti HTML, CSS, JavaScript untuk *frontend*, serta PHP dan MySQL untuk *backend*. Pengembangan akan dilakukan secara iteratif, dengan pengujian berkala untuk memastikan fungsionalitas setiap komponen.

Implementasi akan dilakukan dalam lingkungan terbatas di Bandara NOP Goliat Dekai. Pada tahap ini, aplikasi akan diuji dalam situasi nyata, melibatkan staf bandara dan sampel penumpang. *Feedback* yang diperoleh pada tahap ini akan sangat berharga untuk penyempurnaan aplikasi.

Tahap evaluasi akan menilai efektivitas aplikasi dalam menangani pelaporan kehilangan bagasi. Evaluasi ini akan menggunakan metode campuran (*mixed method),* menggabungkan data kuantitatif (seperti waktu penanganan laporan) dan data kualitatif (seperti kepuasan pengguna). Hasil evaluasi akan menjadi dasar untuk perbaikan dan pengembangan lebih lanjut.

Penggunaan metode R&D dalam penelitian ini diharapkan dapat

menghasilkan aplikasi berbasis web yang tidak hanya fungsional, tetapi juga benar-benar menjawab kebutuhan spesifik Bandara NOP Goliat Dekai dalam penanganan pelaporan kehilangan bagasi. Melalui proses yang sistematis dan iteratif, penelitian ini bertujuan untuk menghasilkan solusi teknologi yang dapat meningkatkan efisiensi operasional bandara dan kepuasan penumpang.Waktu penelitian.

B. Waktu dan Tempat Penelitian

Penelitian ini dilaksanakan di Bandara Nop Goliat Dekai, yang terletak di Kabupaten Yahukimo, Provinsi Papua Pegunungan. Pemilihan lokasi ini didasarkan pada relevansi topik penelitian dengan aktivitas yang terjadi di lingkungan bandara, serta aksesibilitas data dan informan yang diperlukan dalam proses pengumpulan data.

Adapun waktu pelaksanaan penelitian ini berlangsung selama .± 2 (dua) bulan di tahun 2024.

C. Alat dan bahan penelitian

Dalam rangka melaksanakan penelitian "Aplikasi Berbasis Web Untuk Penanganan Pelaporan Kehilangan Bagasi Di Bandara Nop Goliat Dekai", diperlukan berbagai alat dan bahan yang mendukung proses pengembangan, implementasi, dan evaluasi. Alat dan bahan ini dipilih dengan mempertimbangkan kebutuhan spesifik penelitian, kondisi dilokasi, serta standar teknologi terkini dalam pengembangan aplikasi web.

Alat penelitian yang digunakan meliputi perangkat keras (*hardware*) dan perangkat lunak (*software*). Perangkat keras utama yang digunakan adalah laptop

dengan spesifikasi yang memadai untuk pengembangan aplikasi web, yaitu prosesor Intel Core i5 generasi ke-11, RAM 16 GB, dan penyimpanan SSD 512 GB. Laptop ini dipilih karena kemampuannya menjalankan berbagai software pengembangan dengan lancar dan mendukung multitasking yang diperlukan dalam proses coding dan desain.

Selain laptop, penelitian ini juga menggunakan smartphone berbasis Android dan iOS untuk pengujian responsivitas aplikasi pada perangkat mobile. Penggunaan kedua platform ini penting untuk memastikan kompatibilitas aplikasi dengan berbagai jenis perangkat yang mungkin digunakan oleh staf bandara dan penumpang.

Untuk keperluan dokumentasi dan analisis visual, kamera DSLR Canon EOS 80D digunakan. Kamera ini memiliki kemampuan merekam video berkualitas tinggi, yang berguna untuk mendokumentasikan proses penanganan bagasi yang ada dan interaksi pengguna dengan aplikasi yang dikembangkan.

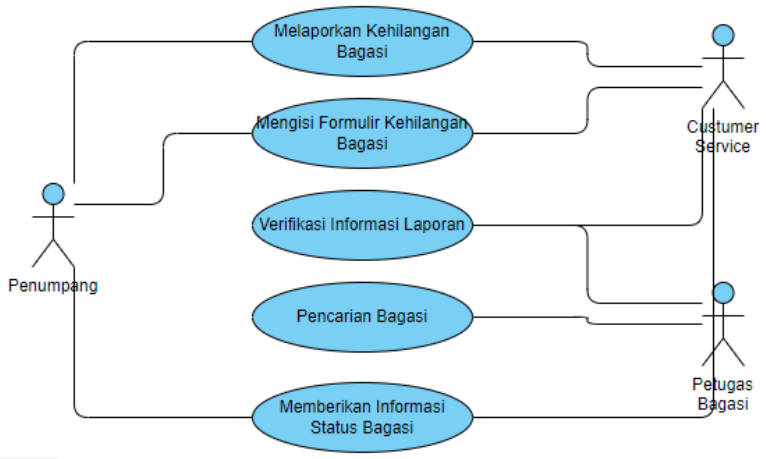
Perangkat lunak yang digunakan dalam penelitian ini mencakup berbagai tools pengembangan dan analisis. Visual Studio Code dipilih sebagai *Integrated Development Environment* (IDE) utama karena fleksibilitasnya dalam mendukung berbagai bahasa pemrograman web. Untuk manajemen basis data, MySQL *Workbench* digunakan sebagai *tool* visualisasi dan *admin*istrasi *database*.

D. Rancangan sistem

1. Sistem yang berjalan

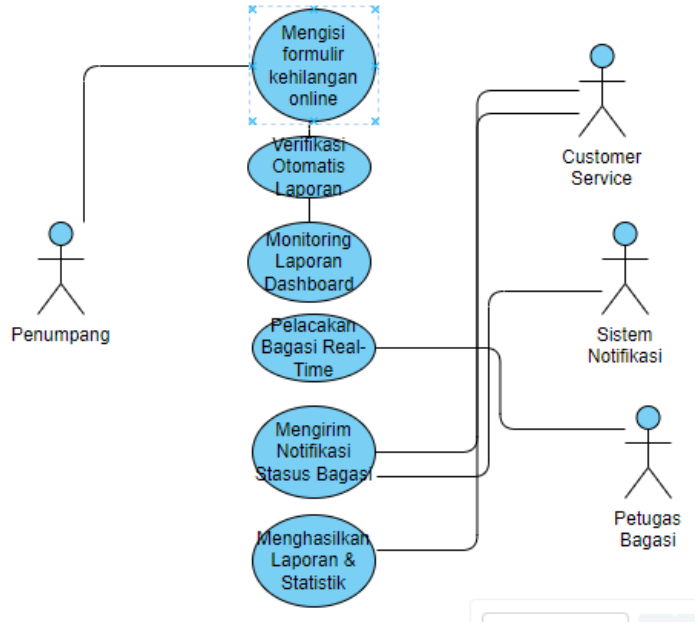
Pada tahap awal sistem yang telah berjalan, proses dimulai dengan cara *User* melakukan pelaporan manual, penumpang yang kehilangan bagasi harus melapor

ke bagian layanan pelanggan di bandara, setelah itu penumpang mengisi formulir secara manual, yang mencakup informasi seperti nomer penerbangan, deskripsi bagasi, waktu kehilangan dan detail kontak, setelah laporan diterima, petugas bandara akan melakukan pencarian dan pelacakan bagasi secara manual dengan bantuan *system internal* atau komunikasi langsung dengan staf lain di bandara. Proses verifikasi data dilakukan oleh petugas bandara melalui *cross-check* dengan sistem bagasi dan informasi penerbangan. Ini melibatkan pemeriksaan terhadap laporan, status penerbangan dan data penumpang. Pelaporan status; petugas kemudian akan memberitahu penumpang tentang status pencarian bagasi mereka melalui telepon atau *email*.



Gambar 3. 1 Sistem yang berjalan

2. Sistem yang diusulkan



Gambar 3. 2 Sistem yang diusulkan

Sistem yang diusulkan dalam penanganan pelaporan kehilangan bagasi di Bandara Nop Goliat Dekai menggambarkan interaksi antara empat aktor utama: Penumpang, *Customer Service*, Petugas Bagasi, dan Sistem Notifikasi.

Penumpang adalah aktor yang mengalami kehilangan bagasi dan menginisiasi proses dengan Mengisi Formulir Kehilangan Bagasi *Online* melalui *portal web.* Setelah formulir diisi, sistem akan secara otomatis melakukan Verifikasi Otomatis Laporan, dimana data yang dimasukkan oleh penumpang diverifikasi dengan data penerbangan dan bagasi yang ada didatabase. Proses verifikasi ini berjalan tanpa campur tangan manusia, sehingga mempercepat waktu respons dan mengurangi kesalahan.

Setelah laporan diverifikasi, *Customer Service* memantau dan mengelola laporan yang masuk melalui Monitoring Laporan di*Dashboard*. Mereka dapat

melihat status dari setiap laporan, memprioritaskan penanganan kasus, dan memberikan pembaruan kepada penumpang. Petugas Bagasi kemudian mengambil alih proses dengan melakukan Pelacakan Bagasi *Real-Text* menggunakan data yang tersedia disistem, seperti informasi lokasi terakhir dari bagasi yang hilang. Hal ini memungkinkan petugas untuk lebih cepat menemukan bagasi yang hilang.

Sistem Notifikasi berperan penting dalam memberikan transparansi dan kepastian kepada penumpang dengan Mengirim Notifikasi Status Bagasi secara otomatis. Penumpang akan menerima informasi mengenai status laporan mereka, mulai dari konfirmasi penerimaan laporan hingga hasil akhir pencarian bagasi. Notifikasi ini dapat dikirim melalui *email* atau SMS, tergantung pada preferensi penumpang.

Terakhir, *Customer Service* juga memiliki akses untuk Menghasilkan Laporan dan Statistik terkait kasus kehilangan bagasi. Data ini digunakan untuk analisis dan perbaikan sistem lebih lanjut, memungkinkan bandara untuk meningkatkan layanan dan mengurangi insiden kehilangan bagasi di masa depan.

E. Teknik Pengumpulan Data

1. **Studi pustaka**

   Studi pustaka merupakan strategi pengumpulan data dari berbagai macam sumber literatur seperti, buku, jurnal, *e-book*, youtube, .ataupun website-website yang relevan dengan masalah yang diidentifikasi.

2. **Observasi**

   Observasi merupakan teknik pengumpulan data dengan cara mengamati

secara langsung maupun tidak langsung. Observasi tidak langsung dilakukan dengan cara mengamati objek penelitian melalui media tertentu, seperti video atau foto.

3. **Dokumentasi**

   Pada tahap ini dilakukan mengumpulkan data yang berkaitan dalam bentuk dokumen. Dokumen-dokumen tersebut dapat berupa dokumen tertulis, dokumen elektronik, atau dokumen audiovisual.

BAB IV

HASIL PENELITIAN DAN PEMBAHASAN

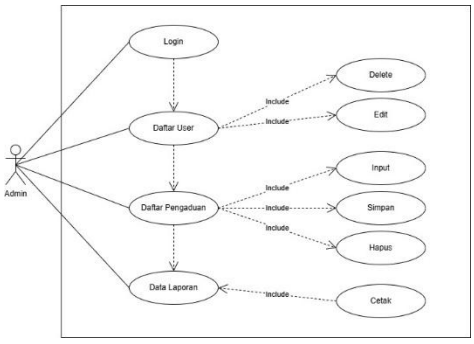
A. Analisis Aliran Data Dengan UML

Dalam analisis sistem ini penulis menggunakan *Use Case* Diagram, *Activity* Diagram dan *Squence* Diagram.

1. *Use Case*

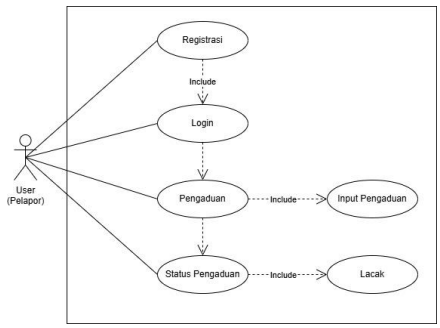
Use Case Diagram berfungsi untuk menjalankan manfaat sistem jika dilihat menurut pandangan orang yang berada diluar sistem (*actor*).

c. *Use Case diagram Admin*



Gambar 4.1 *Use Case Admin*

d. *Use Case diagram User*



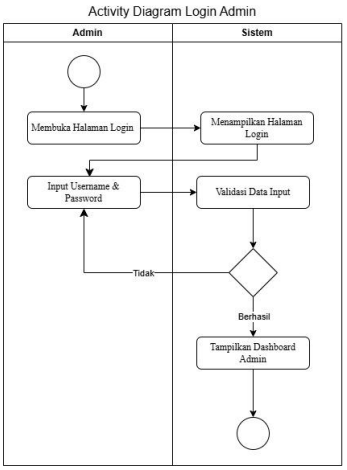
Gambar 4.2 *Use Case User*

2. *Activity Diagram*

Activity diagram adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. *Activity diagram* ini menjelaskan tentang aktifitas-aktifitas yang terjadi didalam sebuah aliran proses pada sebuah *system.*

a. *Activity Diagram Admin*

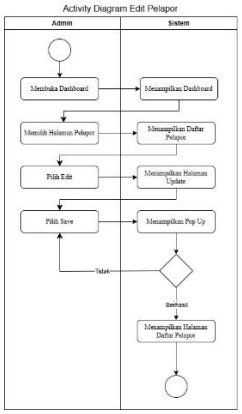
1) *Activity Diagram Login*



Gambar 4.3 *Activity Diagram Login*

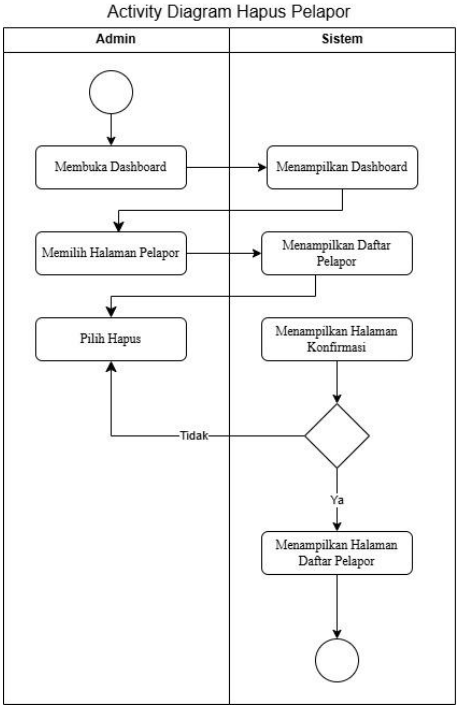
2) *Activity Diagram* Daftar Pelapor

a) *Activity Diagram Edit*



Gambar 4.4 *Activity Diagram* Daftar Pelapor

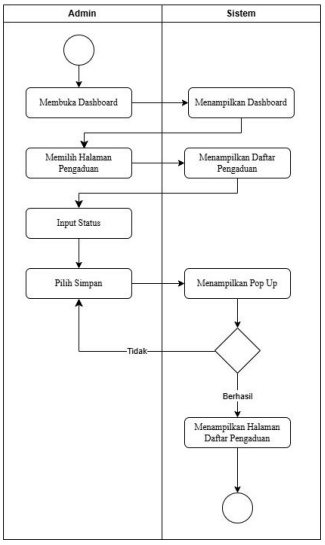
b) *Activity Diagram* Hapus



Gambar 4.5 *Activity Diagram* Hapus

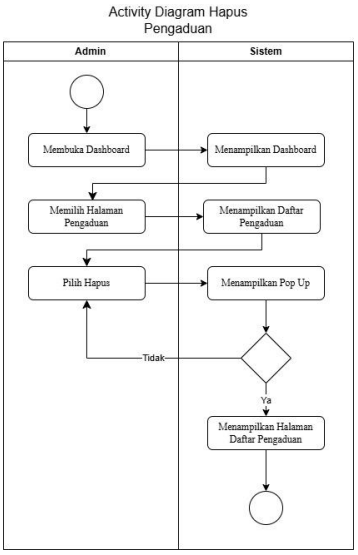
3) *Activity Diagram* Daftar Pengaduan

a) *Activity Diagram* Input Status



Gambar 4.6 *Activity Diagram* Input Status

b) *Activity Diagram* Hapus

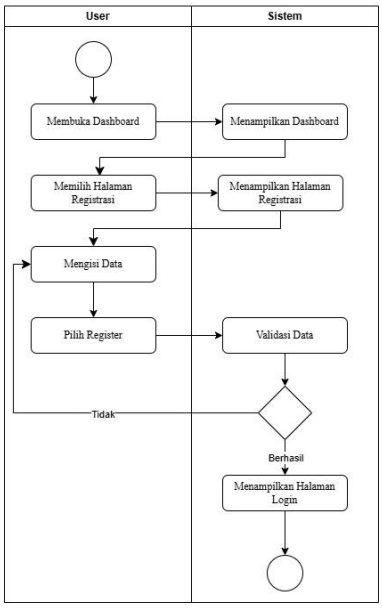


Gambar 4.7 *Activity Diagram* Hapus

4) *Activity Diagram* Data Laporan

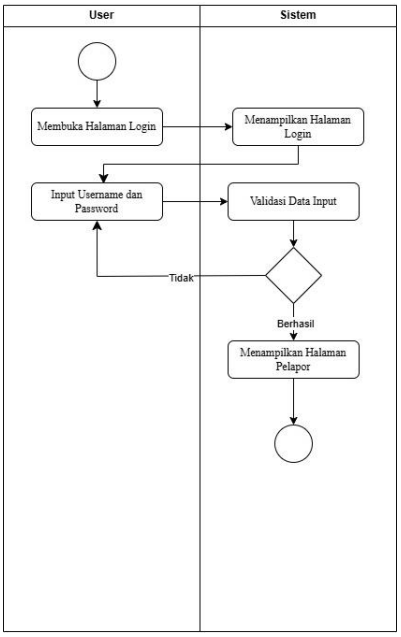
b. *Activity Diagram User*

1) *Activity Diagram Registrasi User*



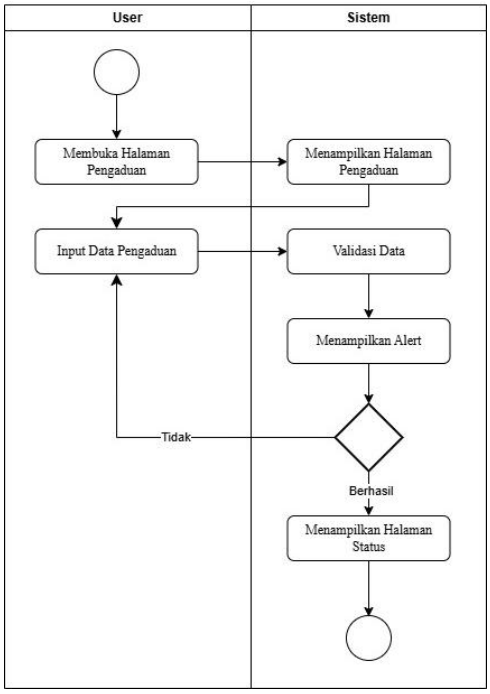
Gambar 4.8 *Activity Diagram Registrasi User*

2) *Activity Diagram Login User*



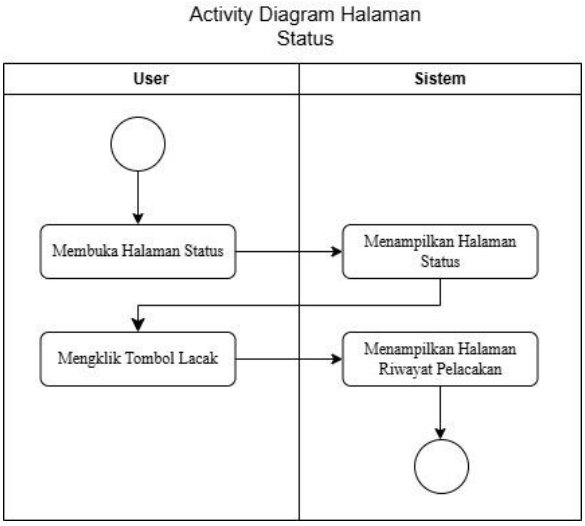
Gambar 4.9 *Activity Diagram Login User*

3) *Activity Diagram* Pengaduan



Gambar 4.10 *Activity Diagram* Pengaduan

4) *Activity Diagram* Status



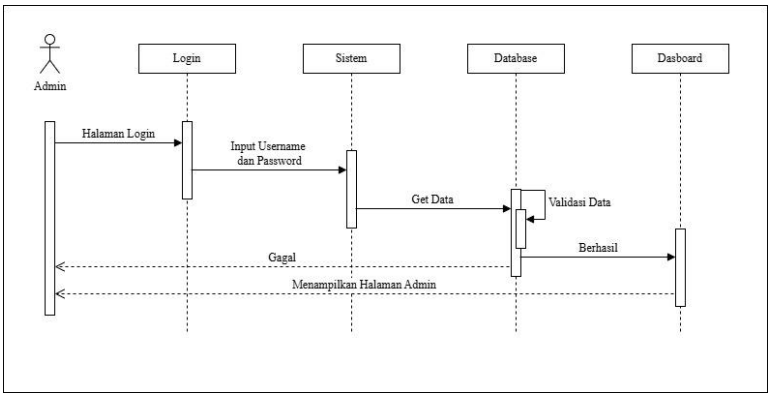
Gambar 4.11 *Activity Diagram* Status

3. Squence Diagram

Squence Diagram merupakan salah satu *Diagram Interaction* yang menjelaskan bagaimana suatu operasi itu dilakukan *message* (pesan) apa yang dikirim dan kapan pelaksanaannya.

a. Squence Diagram Admin

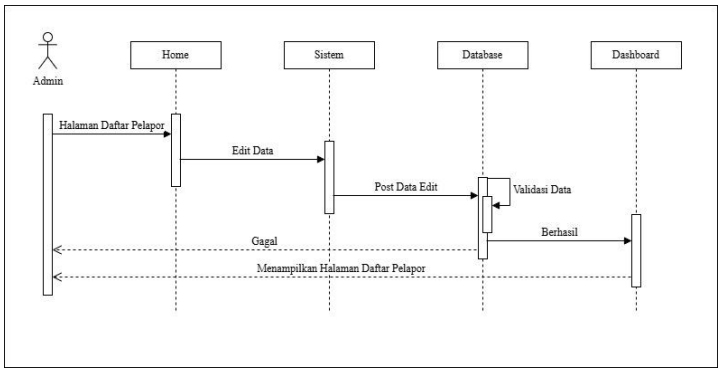
1) Squence Diagram Login Admin



Gambar 4.12 *Squence Diagram Admin*

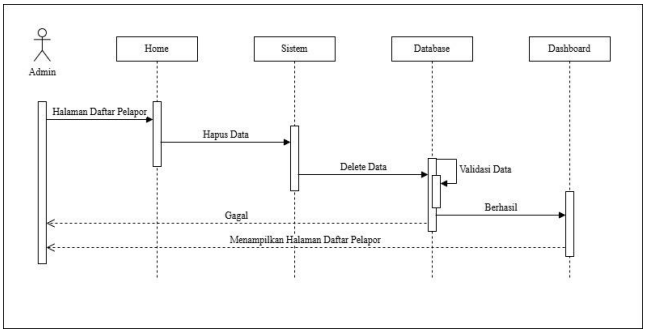
2) *Squence Diagram* Daftar Pelapor

a) *Squence Diagram Edit*



Gambar 4.13 *Squence Diagram Edit*

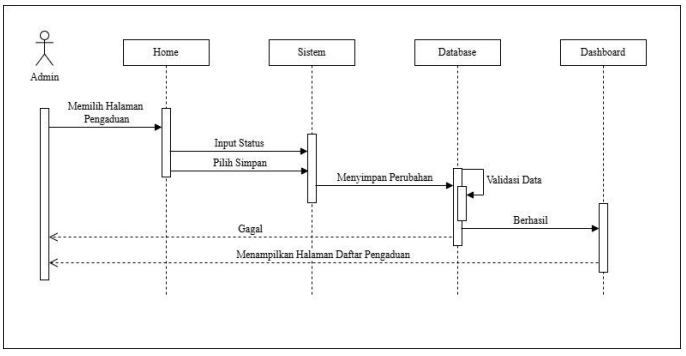
b) *Squence Diagram* Hapus



Gambar 4.14 *Squence Diagram* Hapus

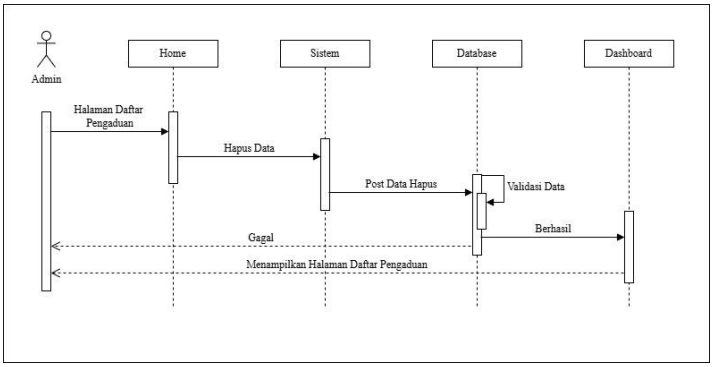
3) *Squence Diagram* Daftar Pengaduan

a) *Squence Diagram Input* Status



Gambar 4.15 *Squence Diagram* Input Status

b) Squence Diagram Hapus

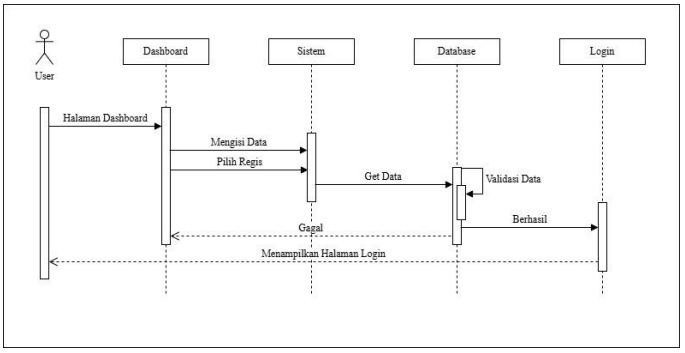


Gambar 4.16 *Squence Diagram* Hapus

4) Squence Diagram Data Laporan

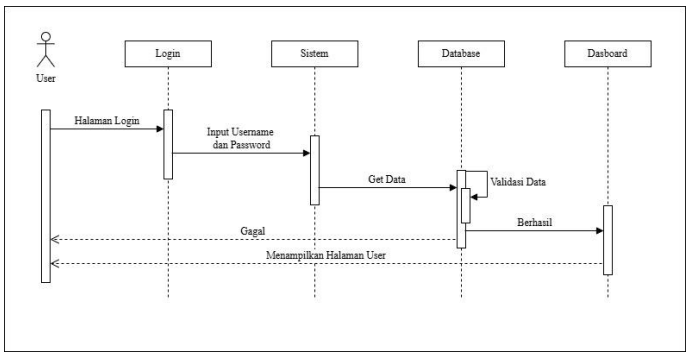
b. Squence Diagram User

1) Squence Diagram Registrasi User



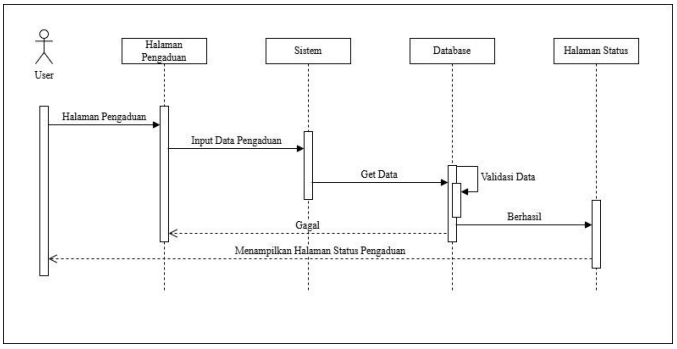
Gambar 4.17 *Squence Diagram Registrasi User*

2) Squence Diagram Login User



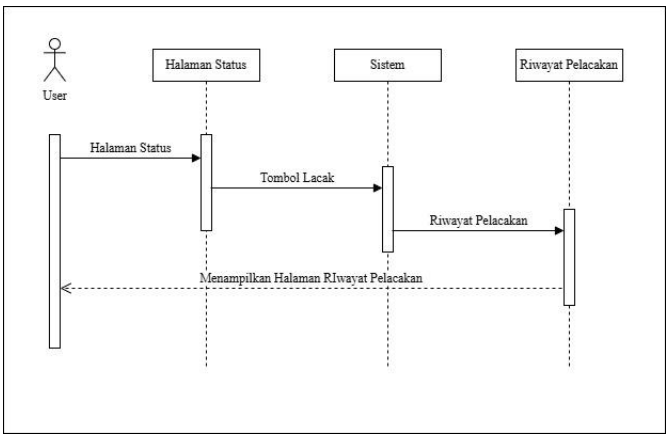
Gambar 4.18 *Squence Diagram Login User*

3) Squence Diagram Pengaduan



Gambar 4.19 *Squence Diagram* Pengaduan

4) Squence Diagram Status



Gambar 4.20 *Squence Diagram* Status

4. Kamus *Data Database*

Kamus data merupakan salah satu komponen penting dari DBMS yang mencankup informasi tentang struktur *database*. Setiap elemen data yang disimpan dalam database memiliki catatan kamus data yang menjelaskan elemen tersebut. Kamus data sangat membantu analisis dalam mendefinisikan dta yang mengalir dalam sistem, sehingga definisi data dapat dilakukan secara lengkap dan terstruktur. Berikut ini adalah kamus data dari aplikasi

Penanganan Pelaporan Kehilangan Bagasi di Bandara NOP Goliat Dekai.

a. Kamus *data admin*

***Tabel* 4. 1** Kamus *data admin*

Nama Atribut	Tipe Data	Ukuran	*Key*
adm_id	int	-	*Primary Key*
Username	vaschar	222	-
password	varchar	222	-
email	varchar	222	-
date	*text*stamp	-	-

Penjelasan kamus *data admin* pada *tabel* sebagai berikut:

1) Amd_id kolom id pengguna dari *tabel admin* sekaligus sebagai *primary key.*
2) *Username* merupakan kolom berisi nama pengguna yang digunakan untuk *login*. Tipe datanya varchar yang memiliki maksimal 222 karakter.
3) *Password* merupakan kolom berisi kata sandi untuk *admin login*. *Password* disimpan dalam bentuk hash (seperti bcrypt), jadi tipe datanya varchar dengan panjang teks 222 karakter.
4) *Email* merupakan kolom yang berisi alamat surel pengguna untuk keperluan komunikasi. *Email* disimpan untuk setiap pengguna agar tidak terjadi duplikasi data. Tipe datanya varchar dengan panjang teks 222 karakter.
5) *Date* merupakan kolom yang berisi informasi tanggal, seperti tanggal

pendaftaran.

b. Kamus *data User*

***Tabel* 4.2** Kamus *data User*

Nama Atribut	*Tipe Data*	Ukuran	*Key*
id	*int*	-	*Primary Key*
Username	*varchar*	50	-
fullname	*varchar*	100	-
email	*varchar*	100	-
phone	*varchar*	15	-
password	*varchar*	255	-
created_at	*textstamp*		-
address	*varchar*	255	-

Penjelasan kamus data *User* pada *tabel* sebagai berikut:

1) Id merupakan kolom id pengguna dari *tabel User* sekaligus sebagai *primary key.*
2) *Username* merupakan kolom berisi nama pengguna yang digunakan untuk *login*. Tipe datanya varchar yang memiliki maksimal 50 karakter.
3) *Fullname* merupakan kolom yang berisi nama lengkap pengguna. Tipe datanya varchar dengan panjang teks 100 karakter untuk menampung berbagai jenis variasi nama lengkap.
4) *Email* merupakan kolom yang berisi alamat surel pengguna untuk menyimpan menggunakan tipe data varchar untuk menyimpan teks

dengan panjang variabel 100 karakter.

5) *Phone* merupakan kolom yang berisi nomor telpon pengguna untuk keperluan komunikasi. Tipe datanya varchar yang memiliki maksimal 15 karakter.
6) *Password* merupakan kolom berisi kata sandi untuk *admin login*. *Password* disimpan dalam bentuk hash (seperti bcrypt), jadi tipe datanya varchar dengan panjang teks 255 karakter.
7) *Cerated_at* merupakan kolom yang berisi informasi waktu saat data pertama kali dibuat memiliki tipe data *textstamp*.
8) *Address* merupakan kolom yang berisi informasi alamat pengguna. Kolom ini menggunakan tipe data varchar dengan panjang teks 255 karakter.

c. Kamus data status *history*

***Tabel* 4.3** Kamus data *status history*

Nama Atribut	Tipe Data	Ukuran	*Key*
id	int	-	*Primary Key*
report_id	int	-	-
status	*text*	-	-
update_at	*text*stamp	-	-

Penjelasan kamus data *status_history* pada *tabel* sebagai berikut:

1) Id merupakan kolom id pengguna dari *tabel* status_history sekaligus sebagai *primary key.*
2) *Report_id* merupakan kolom yang berisi identifikasi laporan dalam

sistem yang memiliki tipe data int.

3) *Status* merupakan kolom yang berisi informasi progres dengan menggunakan tipe data *text*.
4) *Update_at* merupakan kolom yang berisi informasi waktu terakhir di perbaharui data yang menggunakan tipe data *textstamp*.

d. Kamus data *notification*

***Tabel* 4.4** *data notification*

Nama Atribut	Tipe Data	Ukuran	*Key*
id	int	-	*Primary Key*
report_id	int	-	-
message	*text*	-	-
status	enum('unread', 'read')	-	-
created_at	*text*stamp	-	-

Penjelasan kamus data *notification* pada *tabel* sebagai berikut:

1) Merupakan kolom id pengguna dari *tabel notification* sekaligus sebagai *primary key.*
2) *Report_id* merupakan kolom yang berisi identifikasi laporan dalam sistem yang memiliki tipe data int.
3) *Message* merupakan kolom berisi pesan yang dikirimkan, menggunakan tipe data *text* digunakan untuk menyimpan teks panjang.
4) *Status* merupakan kolom yang berisi informasi progres dengan menggunakan tipe data enum.

5) *Create_at* merupakan kolom berisi waktu dan tanggal ketika pesan di buat, dengan tipe data *textstamp* karena diisi otomatis saat pembuatan.

e. Kamus data *reports*

***Tabel* 4.5** *data reports*

Nama Atribut	Tipe Data	Ukuran	*Key*
id	int	-	*Primary Key*
*User*_id	int	-	
name	varchar	255	-
contac	varchar	50	-
email	varchar	255	-
flight_number	varchar	50	-
route	varchar	255	-
baggage_descruption	*text*	-	-
date_lost	date	-	-
*text*_lost	*text*	-	-
photo	varchar	255	-
status	varchar	255	-
created_at	*text*stamp	-	-
update_at	*text*stamp	-	-

Penjelasan kamus data *reports* pada *tabel* sebagai berikut:

1) Id merupakan kolom id pengguna dari *tabel reports* sekaligus sebagai *primary key*.

2) *User_id* merupakan kolom yang berisi identifikasi untuk setiap

pengguna, memiliki tipe data int.

3) *Name* merupakan kolom yang berisi nama pengguna, kolom ini digunakan untuk menyimpan nama pelapor yang memiliki tipe data varchar dengan panjang teks maksimal 255.
4) *Contac* merupakan kolom yang berisi informasi nomor telepon pengguna dari *tabel* reports. Kolom ini menggunakan tipe data varchar dengan panjang teks maksimal 50.
5) *Email* merupakan kolom yang menyimpan *email* pelapor yang dapat digunakan untuk komunikasi. Tipe datanya varchar dengan panjang maksimal 255 karakter.
6) *Filght_number* merupakan kolom yang berisi nomor penerbangan, tipe datanya varchar dengan panjang teks maksimal 50 karakter.
7) *Route* merupakan kolom yang dapat menyimpan data asal dan tujuan pelapor. Tipe datanya varchar dengan panjang teks 255.
8) *Baggage_descruption* kolom ini digunakan untuk menyimpan informasi tentang karakteristik bagasi yang dilaporkan dengan tipe data *text* agar dapat menampung deskripsi dengan panjang.
9) *Date_lost* merupakan kolom yang berisi informasi tanggal kehilangan bagasi, tipe datanya date.
10) *Text_lost* merupakan kolom yang berisi infortmasi waktu kehilangan bagasi, tipe datanya *text*.
11) *Photo* merupakan kolom yang berisi *file* gambar bagasi. Kolom ini digunakan untuk menyimpan *path file* foto bagasi yang di unggah oleh

pelapor. Tipe datanya varchar dengan panjang teks maksimal 255 karakter.

12) *Status* kolom ini digunakan untuk mencatat status terkini pada laporan, tipe datanya varchar dengan panjang teks 255 karakter.

13) *Created_at* kolom ini berisi informasi waktu saat laporan dibuat. Tipe datanya *text*stamp.

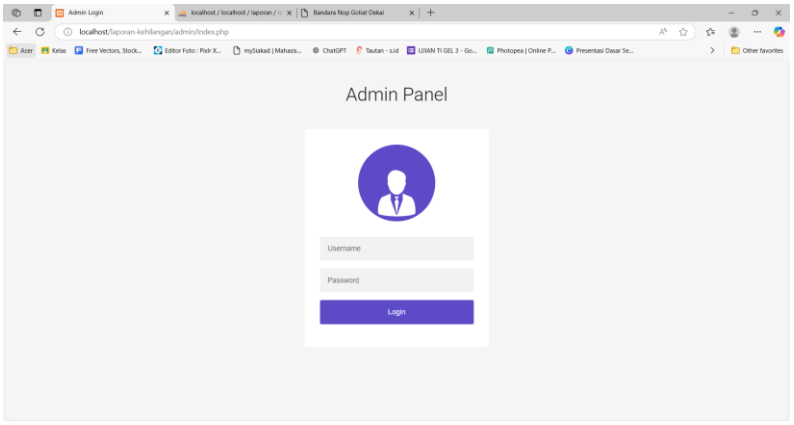
14) *Update_at* kolom ini berisi informasi waktu terakhir laporan diperbaharui. Tipe datanya *text*stamp.

B. Detail Sistem

1. *Admin*

a. Halaman *Login Admin*

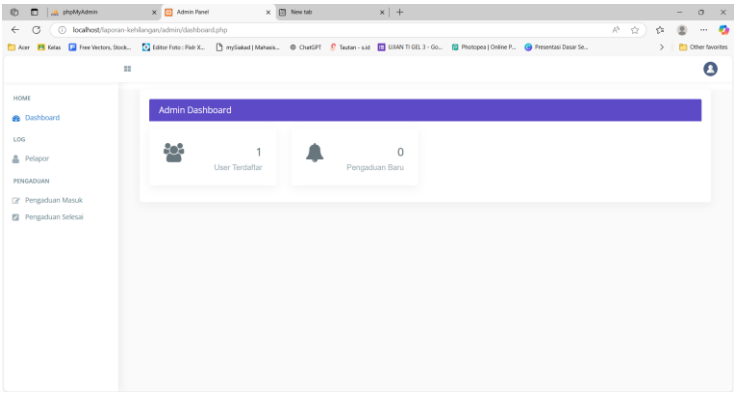
Merupakan halaman awal yang digunakan *admin* untuk mendapatkan akses masuk ke halaman *dashboard admin*. Hanya *admin*istrator yang berwenang yang dapat mengakses dan mengelola data dan fungsi sistem, halaman *login* ini yang berfungsi sebagai pintu gerbang bagi *admin*istrator untuk masuk ke sistem setelah memverifikasi identitasnya.



Gambar 4.21 Halaman *Login Admin*

b. Halaman *Dashboard Admin*

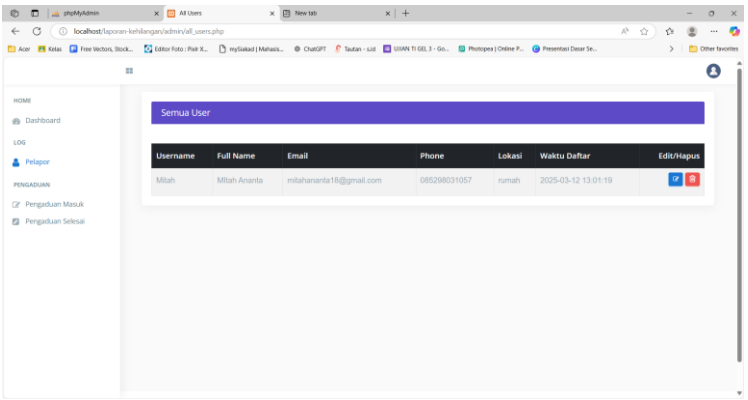
Merupakan halaman utama setelah *admin* berhasil *login*, yang menampilkan daftar pengguna terdaftar, laporan pengaduan yang masuk, dan pengaduan yang telas selesai diproses.



Gambar 4.22 *Dashboard Admin*

c. Halaman Pelapor

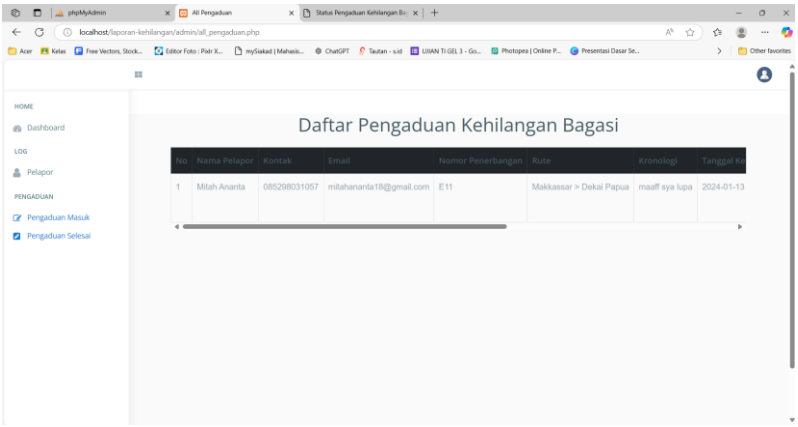
Halaman pelapor adalah bagian dari sistem yang dirancang khusus untuk memungkinkan *admin* mengelola dan melihat semua pengguna yang terdaftar.



Gambar 4.23 Halaman Pelapor

d. Halamn Pengaduan Masuk

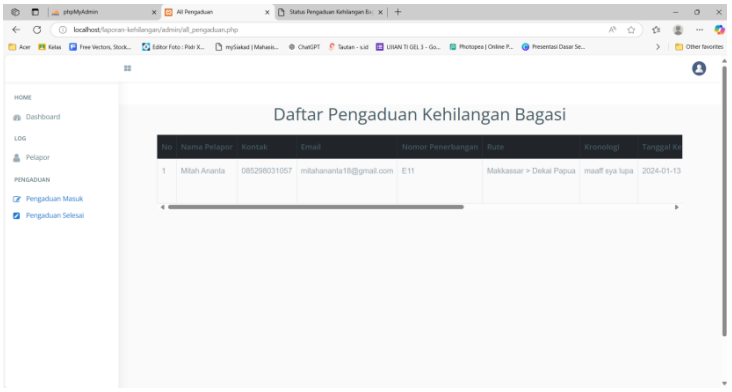
Merupakan bagian dari sistem yang dirancang khusus untuk memungkinkan *admin* melihat dan mengelola laporan kehilangan bagasi yang diajukan oleh pengguna, termasuk meninjau detail laporan dan mengambil tindakan lanjutan.



Gambar 4.24 Halaman Pengaduan Masuk

e. Halaman Pengaduan Selesai

Halaman pengaduan dirancang untuk memungkinkan *admin* melihat dan mengelola laporan kehilangan bagasi yang telah diproses dan diselesaikan.

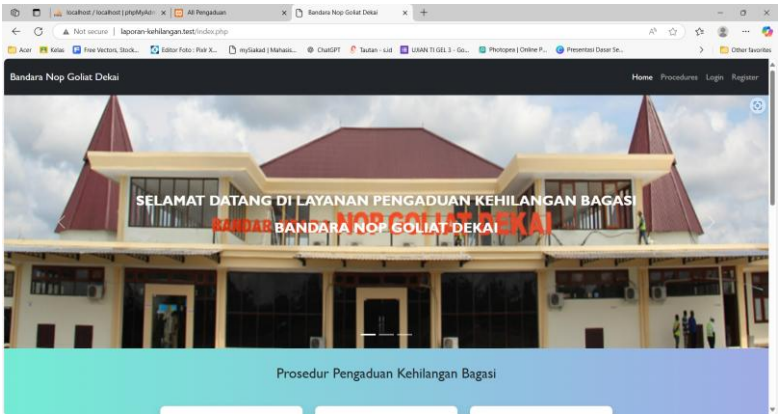


Gambar 4.25 Halaman Pengaduan Selesai

2. *User*

a. Halaman *Home*

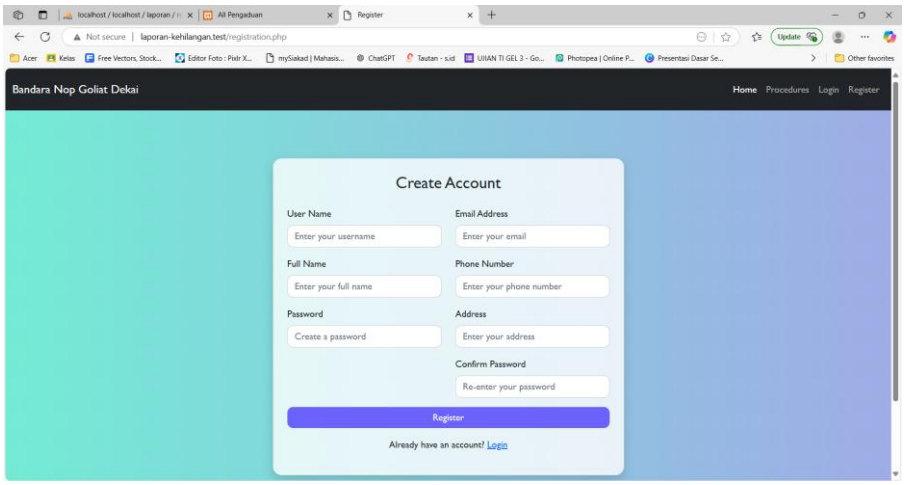
Halaman *Home* adalah bagian dari sistem yang dirancang khusus untuk pengguna, menampilkan informasi utama, panduan pelaporan kehilangan bagasi, serta akses cepat ke fitur pengajuan laporan.



Gambar 4.26 Halaman *Home*

b. Halaman *Register*

Halaman *Register* adalah bagian dari sistem yang dirancang khusus untuk memungkinkan pengguna baru mendaftarkan akun dengan mengisi informasi seperti nama, *email*, nomor telepon, dan kata sandi agar dapat mengakses fitur pelaporan kehilangan bagasi.

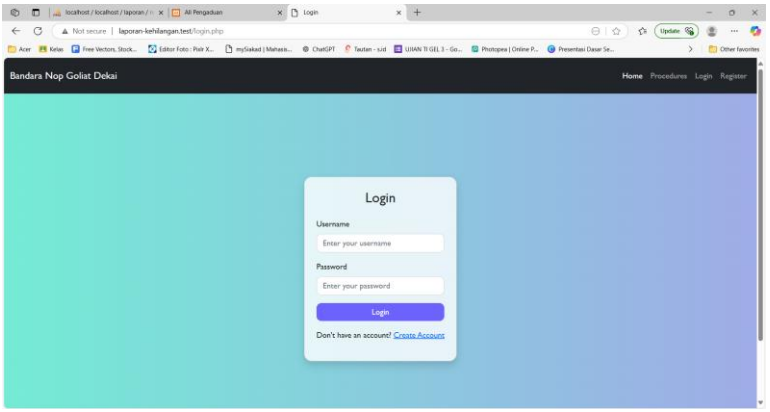


Gambar 4.27 Halaman *Register*

c. Halaman *Login User*

Halaman *login User* adalah bagian dari sistem yang dirancang khusus untuk memungkinkan pengguna mengakses akun mereka dengan memasukkan *email* dan kata sandi yang valid agar dapat menggunakan fitur pelaporan

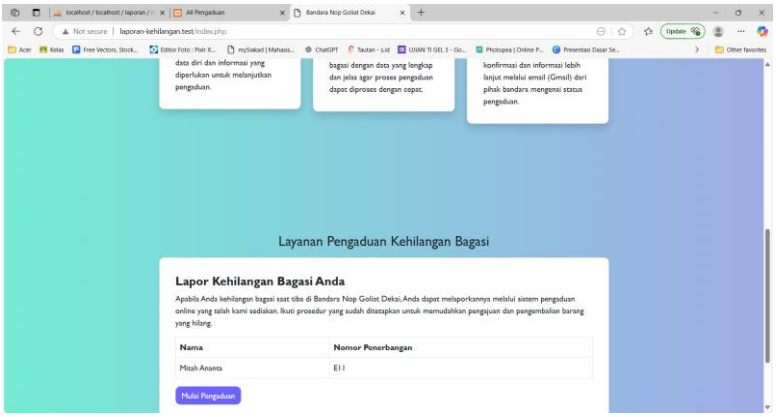
dan memantau status laporan.



Gambar 4.28 Halaman *Login User*

d. Halaman *Home User*

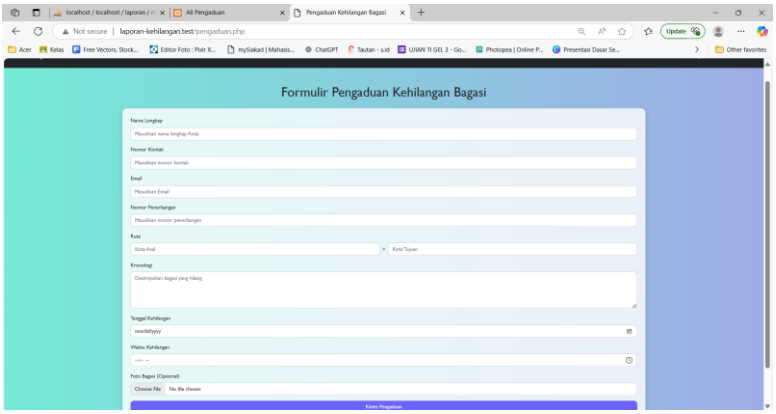
Halaman *Home User* adalah bagian dari sistem yang dirancang khusus untuk menampilkan informasi utama bagi pengguna, termasuk panduan pelaporan kehilangan bagasi, status laporan terbaru, serta akses cepat ke fitur pelaporan dan riwayat pengaduan.



Gambar 4.29 Halaman *Home User*

e. Halaman pengaduan

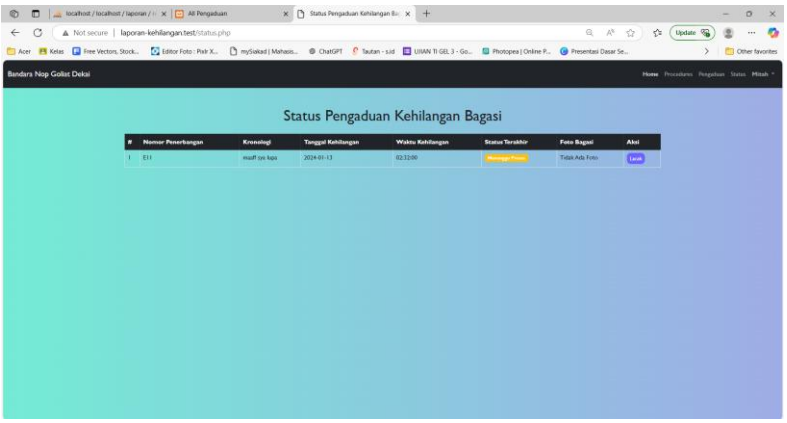
Halaman pengaduan adalah bagian dari sistem yang dirancang khusus untuk memungkinkan pengguna mengajukan laporan kahilangan bagasi dengan mengisi detail informasi.



Gambar 4.30 Halaman Pengaduan

f. Halaman Status

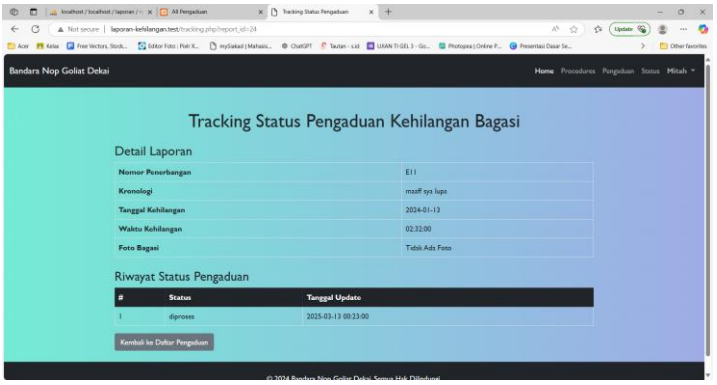
Halaman status adalah bagian dari sistem yang dirancang khusus untuk memungkinkan pengguna memantau perkembangan laporan kehilangan bagasi.



Gambar 4.31 Halaman Status

g. Halaman Lacak

Halaman lacak adalah bagian dari sistem yang dirancang khusus untuk memungkinkan pengguna melacak perkembangan laporan kehilangan bagasi secara real-*text*, melihat riwayat penanganan, serta mendapatkan informasi terbaru terkait status pencarian bagasi.



Gambar 4.32 Halaman Lacak

C. Pengujian Sistem

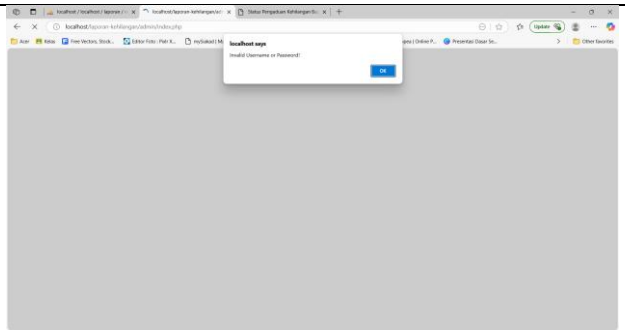
Pengujian sistem pada penelitian ini menggunakan 2 metode yaitu *Black Box Testing* dan *White Box Testing.* Berikut ini hasil pengujian menggunakan sistem tersebut.

1. *Black Box Testing*

   a. *Balck Box Testing* Kesalahan *Username* dan *Password*

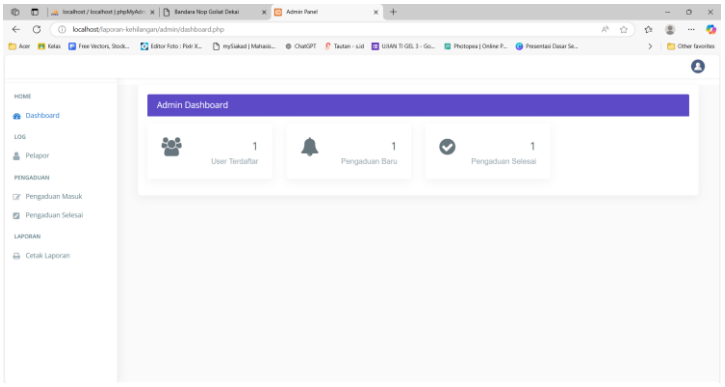
***Tabel* 4.6** *Black Box Testing*

Tes Faktor	Hasil	Keterangan
Memasukkan *Username* atau *Password* yang tidak sesuai	✓	Berhasil, ketika *Username* atau *password* tidak sesuai maka tampil pesan Invalid *Username* or *Password*!
Screenshot		



b. *Black Box Testing Admin* Berhasil *Login*

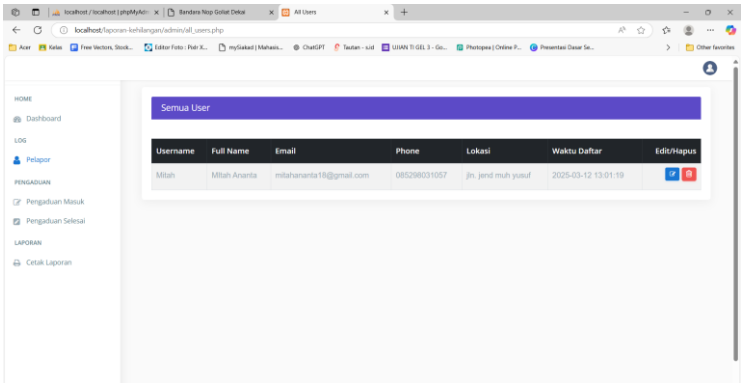
Tabel 4.7 *Black Box Testing Admin* Berhasil *Login*

Tes Faktor	Hasil	Keterangan
Memasukkan *Username* atau *Password* yang tidak sesuai	✓	Berhasil, sistem menampilkan halaman *dashboard Admin*
Screenshot		
		

c. *Black Box Testing Edit* Pelapor

Tabel 4.8 *Black Box Testing Edit* Pelapor

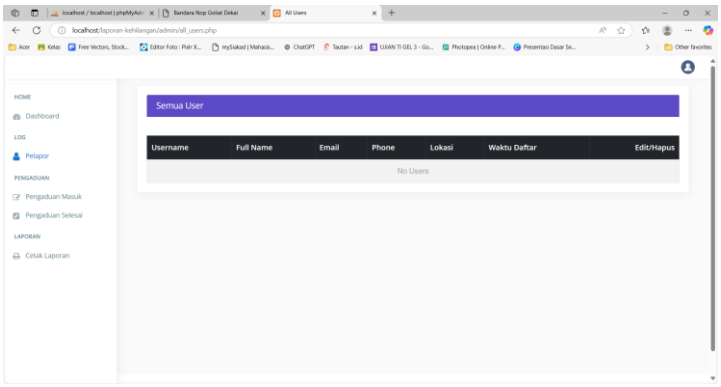
Tes Faktor	Hasil	Keterangan
Admin menekan tombol *edit* kemudian memperbaharui data dan menekan tombol *save*	✓	Berhasil, sistem memperbaharui data dan menampilkan halaman Pelapor
Screenshot		



d. *Black Box Testing* Hapus Pelapor

Tabel 4.9 *Black Box Testing* Hapus Pelapor

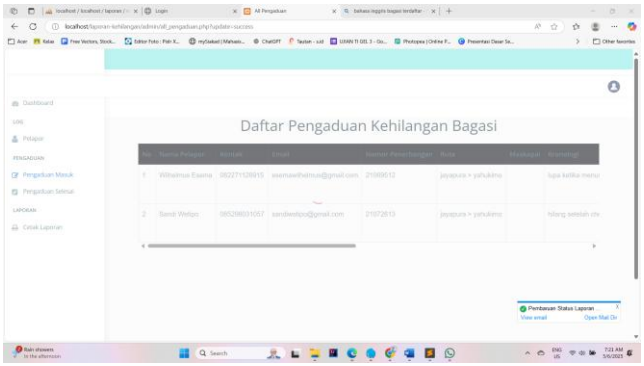
Tes Faktor	Hasil	Keterangan
Admin menekan tombol hapus dan mengkonfirmasi penghapusan data	✓	Berhasil, sistem menghapus data dan memperbarui tampilan halaman Pelapor
Screenshot		



e. Black Box Testing Simpan Pengaduan Masuk

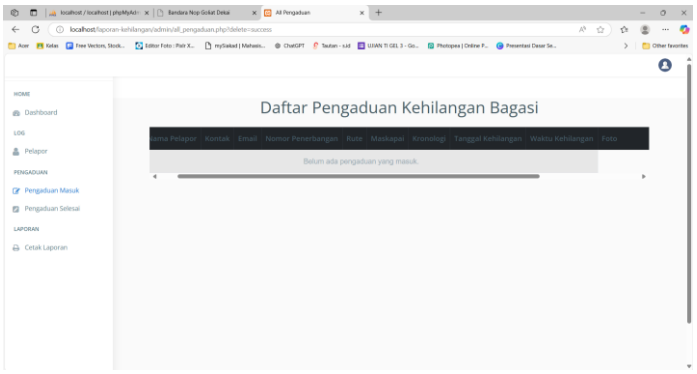
Tabel 4.10 *Black Box Testing* Simpan Pengaduan Masuk

Tes Faktor	Hasil	Keterangan
Admin menekan tombol simpan untuk	✓	Berhasil, sistem memperbarui status pengaduan

memperbarui status pengaduan		dan menampilkan notifikasi sukses
Screenshot		
		

f. *Black Box Testing* Hapus Pengaduan Masuk

Tabel 4.11 *Black Box Testing* Hapus Pengaduan Masuk

Tes Faktor	Hasil	Keterangan
Admin menekan tombol hapus dan mengkonfirmasi penghapusan data	✓	Berhasil, sistem menghapus data dan memperbarui tampilan
Screenshot		
		

g. *Black Box Testing* Cetak Laporan

***Tabel* 4.12** *Black Box Testing* Cetak Laporan

Tes Faktor	Hasil	Keterangan
Admin memasukkan	✓	Berhasil, sistem menampilkan

rentang tanggal untuk mencari laporan yang akan dicetak, lalu menekan tombol Cetak		laporan sesuai dengan tanggal yang dipilih dan menampilkan opsi cetak
Screenshot		
		

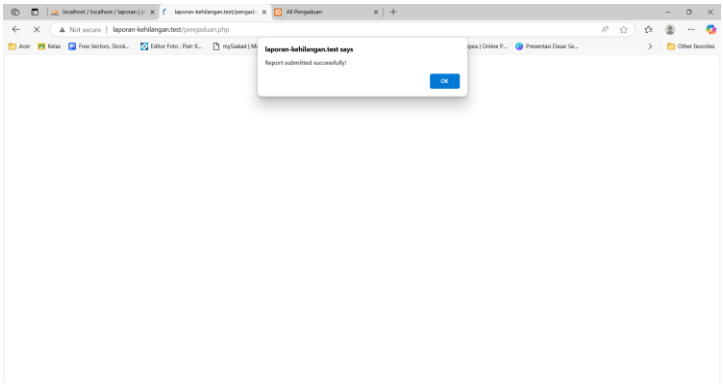
h. *Black Box Testing* Halaman *Dashboard User*

***Tabel* 4.13** *Black Box Testing* Halaman *Dashboard User*

Tes Faktor	Hasil	Keterangan
User membuka halaman *dashboard*	✓	Berhasil, sistem menampilkan halaman *dashboard*
Screenshot		
		

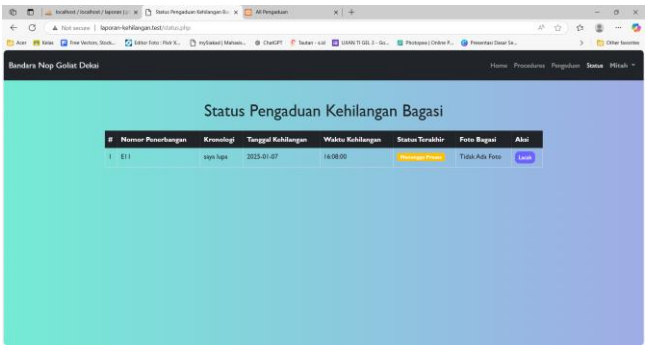
i. *Black Box Testing* Halaman Pengaduan

***Tabel* 4.14** *Black Box Testing* Halaman Pengaduan

Tes Faktor	Hasil	Keterangan
User mengakses halaman pengaduan, mengisi formulir pengaduan dan menekan tombol kirim pengaduan	✓	Berhasil, sistem menyimpan pengaduan dan menampilkan notifikasi sukses
Screenshot		
		

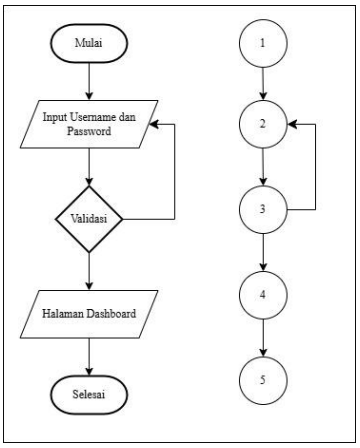
j. *Black Box Testing* Halaman Status

***Tabel* 4.15** *Black Box Testing* Halaman Status

Tes Faktor	Hasil	Keterangan
User mengecek status pengaduan yang telah dikirim	✓	Berhasil, sistem menampilkan status pengaduan
Screenshot		
		

2. *White Box Testing*

a) *White Box Testing* Kesalahan *Username* dan *Password*



Gambar 4.33 *Flowchart* dan *Flowgraph* Kesalahan *Username* dan *Password*

Perhitungan berikut dilakukan menggunakan gambar di atas:

(1) Tentukan kompleksitas siklomatik node dan tepi V(G).

Pada persamaan: E (tepi) = 5 N (simpul) = 5 P (simpul predikat) = 1 V(G) = E – N + 2

Menjawab: E – N + 2 = 5 – 5 + 2 = 2 adalah V(G).

Predikat (P) sama dengan P + 1 = 1 + 1 = 2.

(2) Region = 2 merupakan hasil perhitungan *Cyclomatic Complexity* dari *flowgraph* diatas.

(3) Jalur 1 = 1 – 2 – 3 – 2 merupakan jalur *independen flowgraph*.

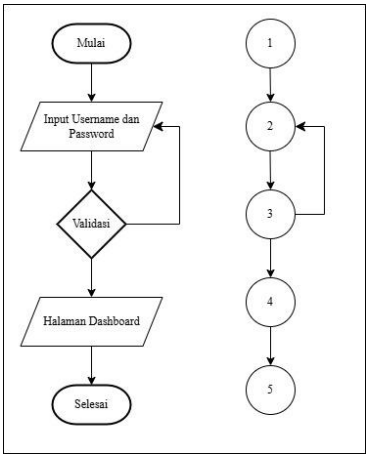
Jalur 2 sama dengan 1–2–3–4–5.

***Tabel* 4.16** Grafik matriks Kesalahan *Username* dan *Password*

	1	2	3	4	5	E-1
1		1				1 – 1 = 0

2			1			1 – 1 = 0
3		1		1		2 – 1 = 1
4					1	1 – 1 = 0
5						0
	SUM (E + 1)					1 + 1 = 2

b) *White Box Testing Admin* Berhasil *Login*



Gambar 4.34 *Flowchart* dan *Flowgraph Admin* Berhasil

Perhitungan berikut dilakukan menggunakan gambar di atas:

(1) Tentukan kompleksitas siklomatik node dan tepi V(G).

Pada persamaan: E (tepi) = 8, N (simpul) = 8, P (simpul predikat) = 1 V(G) = E – N + 2

Menjawab: E – N + 2 = 5 – 5 + 2 = 2 adalah V(G).

Predikat (P) sama dengan P + 1 = 1 + 1 = 2.

(2) Region = 2 merupakan hasil perhitungan *Cyclomatic Complexity* dari *flowgraph* diatas.

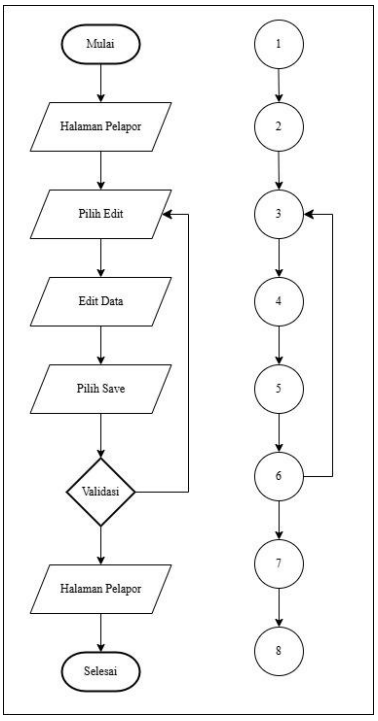
(3) Jalur 1 = 1 – 2 – 3 – 2 merupakan jalur *independen flowgraph*.

Jalur 2 sama dengan 1–2–3–4–5.

***Tabel* 4.17** Grafik matriks *Admin* Berhasil *Login*

	1	2	3	4	5	E-1
1		1				1 – 1 = 0
2			1			1 – 1 = 0
3		1		1		2 – 1 = 1
4					1	1 – 1 = 0
5						0
	SUM (E + 1)					1 + 1 = 2

c) *White Box Testing Edit* Pelapor



Gambar 4.35 *Flowchart* dan *Flowgraph Edit* Pelapor

Perhitungan berikut dilakukan menggunakan gambar di atas:

(1) Tentukan kompleksitas siklomatik node dan tepi V(G).

Pada persamaan: E (tepi) = 5, N (simpul) = 5, P (simpul predikat) = 1 V(G) = E – N + 2

Menjawab: E – N + 2 = 8 – 8 + 2 = 2 adalah V(G).

Predikat (P) sama dengan P + 1 = 1 + 1 = 2.

(2) Region = 2 merupakan hasil perhitungan *Cyclomatic Complexity* dari *flowgraph* diatas.

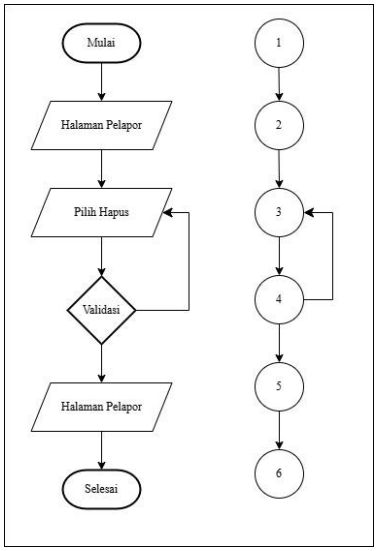
(3) Jalur 1 = 1–2–3–4–5–6–7–8 merupakan jalur *independen flowgraph*.

Jalur 2 = 1–2–3–4–5–6–3– 4–5–6–7–8.

***Tabel* 4.18** *Grafik matriks Edit* Pelapor

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	E-1
1	0	1	0	0	0	0	0	0	1-1=0
2	0	0	1	0	0	0	0	0	1-1=0
3	0	0	0	1	0	0	0	0	1-1=0
4	0	0	0	0	1	0	0	0	1-1=0
5	0	0	0	0	0	1	0	0	1-1=0
6	0	0	1	0	0	0	1	0	2-1=1
7	0	0	0	0	0	0	0	1	1-1=0
8	0	0	0	0	0	0	0	0	0
	SUM (E + 1)								1+1=2

d) *White Box Testing* Hapus Pelapor



Gambar 4.36 *Flowchart* dan *Flowgraph* Hapus Pelapor

Perhitungan berikut dilakukan menggunakan gambar di atas:

(1) Tentukan kompleksitas siklomatik node dan tepi V(G).

Pada persamaan: E (tepi) = 6, N (simpul) = 6, P (simpul predikat) = 1 V(G) = E – N + 2

Menjawab: E – N + 2 = 6 – 6 + 2 = 2 adalah V(G).

Predikat (P) sama dengan P + 1 = 1 + 1 = 2.

(2) Region = 2 merupakan hasil perhitungan *Cyclomatic Complexity* dari *flowgraph* diatas.

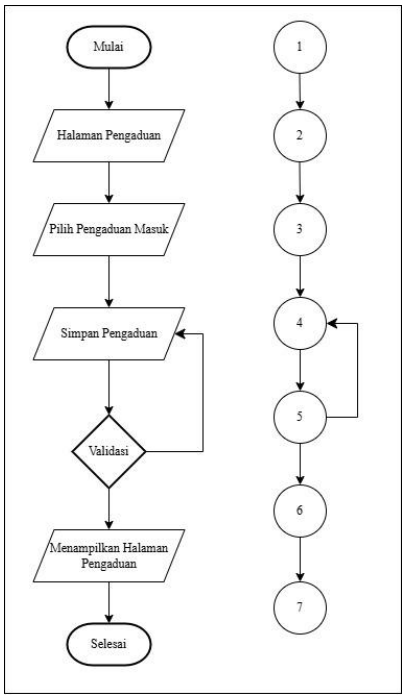
(3) Jalur 1 = 1–2–3–4–5–6 merupakan jalur *independen flowgraph*.

Jalur 2 = 1–2–3–4–3–4–5–6.

***Tabel* 4.19** *Grafik matriks* Hapus Pelapor

	1	**2**	**3**	**4**	**5**	**6**	E-1
1	0	1	0	0	0	0	1-1=0
2	0	0	1	0	0	0	1-1=0
3	0	0	0	1	0	0	1-1=0
4	0	0	1	0	1	0	2-1=1
5	0	0	0	0	0	1	1-1=0
6	0	0	0	0	0	0	0
	SUM (E + 1)						1+1=2

e) *White Box Testing* Simpan Pengaduan Masuk



Gambar 4.37 *Flowchart* dan *Flowgraph* Simpan Pengaduan Masuk

Perhitungan berikut dilakukan menggunakan gambar di atas:

(1) Tentukan kompleksitas siklomatik node dan tepi V(G).

Pada persamaan: E (tepi) = 7, N (simpul) = 7, P (simpul predikat) = 1 V(G) = E – N + 2

Menjawab: E – N + 2 = 7 – 7 + 2 = 2 adalah V(G).

Predikat (P) sama dengan P + 1 = 1 + 1 = 2.

(2) Region = 2 merupakan hasil perhitungan *Cyclomatic Complexity* dari *flowgraph* diatas.

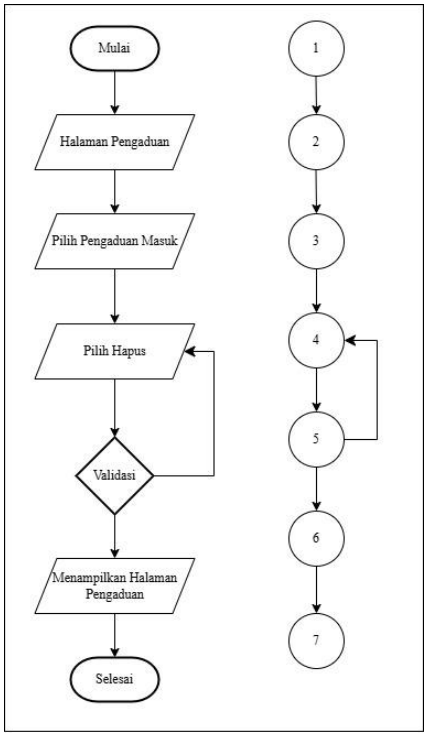
(3) Jalur 1 = 1–2–3–4–5–6–7 merupakan jalur *independen flowgraph*.

Jalur 2 = 1–2–3–4–5–4–5–6–7.

***Tabel* 4.20** *Grafik matriks* Simpan Pengaduan Masuk

	1	**2**	**3**	**4**	**5**	**6**	**7**	E-1
1	0	1	0	0	0	0	0	1-1=0
2	0	0	1	0	0	0	0	1-1=0
3	0	0	0	1	0	0	0	1-1=0
4	0	0	0	0	1	0	0	1-1=0
5	0	0	0	1	0	1	0	2-1=1
6	0	0	0	0	0	0	1	1-1=0
7	0	0	0	0	0	0	0	0
	SUM (E + 1)							1+1=2

f) *White Box Testing* Hapus Pengaduan Masuk



Gambar 4.38 *Flowchart* dan *Flowgraph* Hapus Pengaduan Masuk

Perhitungan berikut dilakukan menggunakan gambar di atas:

(1) Tentukan kompleksitas siklomatik node dan tepi V(G).

Pada persamaan: E (tepi) = 7, N (simpul) = 7, P (simpul predikat) = 1 V(G) = E – N + 2

Menjawab: E – N + 2 = 7 – 7 + 2 = 2 adalah V(G).

Predikat (P) sama dengan P + 1 = 1 + 1 = 2.

(2) Region = 2 merupakan hasil perhitungan *Cyclomatic Complexity* dari *flowgraph* diatas.

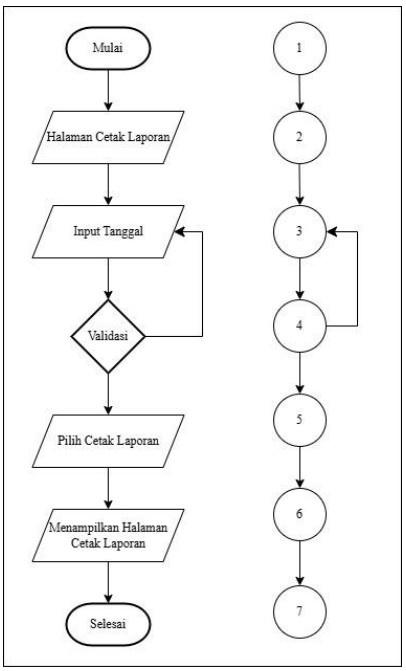
(3) Jalur 1 = 1–2–3–4–5–6–7 merupakan jalur *independen flowgraph*.

Jalur 2 = 1–2–3–4–5–4–5–6–7.

***Tabel* 4.21** *Grafik matriks* Hapus Pengaduan Masuk

	1	**2**	**3**	**4**	**5**	**6**	**7**	E-1
1	0	1	0	0	0	0	0	1-1=0
2	0	0	1	0	0	0	0	1-1=0
3	0	0	0	1	0	0	0	1-1=0
4	0	0	0	0	1	0	0	1-1=0
5	0	0	0	1	0	1	0	2-1=1
6	0	0	0	0	0	0	1	1-1=0
7	0	0	0	0	0	0	0	0
	SUM (E + 1)							1+1=2

g) *White Box Testing* Cetak Laporan



Gambar 4.39 *Flowchart* dan *Flowgraph* Cetak Laporan

Perhitungan berikut dilakukan menggunakan gambar di atas:

(1) Tentukan kompleksitas siklomatik node dan tepi V(G).

Pada persamaan: E (tepi) = 7, N (simpul) = 7, P (simpul predikat) = 1 V(G)

= E – N + 2

Menjawab: E – N + 2 = 7 – 7 + 2 = 2 adalah V(G).

Predikat (P) sama dengan P + 1 = 1 + 1 = 2.

(2) Region = 2 merupakan hasil perhitungan *Cyclomatic Complexity* dari *flowgraph* diatas.

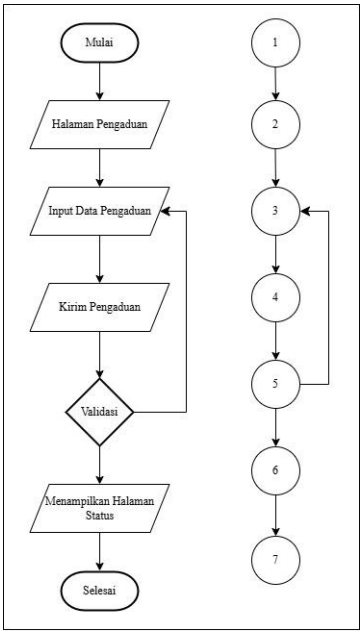
(3) Jalur 1 = 1–2–3–4–5–6–7 merupakan jalur *independen flowgraph*.

Jalur 2 = 1–2–3–4–3–4–5–6–7.

***Tabel* 4.22** *Grafik matriks* Cetak Laporan

	1	**2**	**3**	**4**	**5**	**6**	**7**	E-1
1	0	1	0	0	0	0	0	1-1=0
2	0	0	1	0	0	0	0	1-1=0
3	0	0	0	1	0	0	0	1-1=0
4	0	0	1	0	1	0	0	2-1=0
5	0	0	0	0	0	1	0	1-1=0
6	0	0	0	0	0	0	1	1-1=0
7	0	0	0	0	0	0	0	0
	SUM (E + 1)							1+1=2

h) *White Box Testing* Halaman Pengaduan



Gambar 4.40 *Flowchart* dan *Flowgraph* Halaman Pengaduan

Perhitungan berikut dilakukan menggunakan gambar di atas:

(1) Tentukan kompleksitas siklomatik node dan tepi V(G).

Pada persamaan: E (tepi) = 7, N (simpul) = 7, P (simpul predikat) = 1 V(G) = E – N + 2

Menjawab: E – N + 2 = 7 – 7 + 2 = 2 adalah V(G).

Predikat (P) sama dengan P + 1 = 1 + 1 = 2.

(2) Region = 2 merupakan hasil perhitungan *Cyclomatic Complexity* dari *flowgraph* diatas.

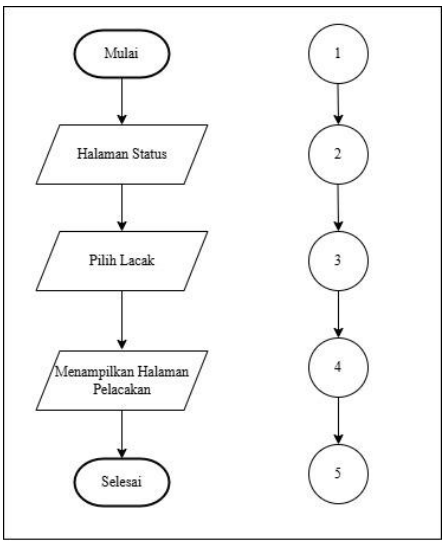
(3) Jalur 1 = 1–2–3–4–5–6–7 merupakan jalur *independen flowgraph*.

Jalur 2 = 1–2–3–4–5–3–4–5–6–7.

***Tabel* 4.23** *Grafik matriks* Halaman Pengaduan

	1	**2**	**3**	**4**	**5**	**6**	**7**	E-1
1	0	1	0	0	0	0	0	1-1=0
2	0	0	1	0	0	0	0	1-1=0
3	0	0	0	1	0	0	0	1-1=0
4	0	0	0	0	1	0	0	1-1=0
5	0	0	1	0	0	1	0	2-1=1
6	0	0	0	0	0	0	1	1-1=0
7	0	0	0	0	0	0	0	0
	SUM (E + 1)							1+1=2

i) *White Box Testing* Halaman Status



Gambar 4.41 *Flowchart* dan *Flowgraph* Halaman Status

Perhitungan berikut dilakukan menggunakan gambar di atas:

(1) Tentukan kompleksitas siklomatik node dan tepi V(G).

Pada persamaan: E (tepi) = 4, N (simpul) = 5, P (simpul predikat) = 1 V(G) = E – N + 2

Menjawab: E – N + 2 = 4 – 5 + 2 = 1 adalah V(G).

Predikat (P) sama dengan P + 1 = 0 + 1 = 1.

(2) Region = 1 merupakan hasil perhitungan *Cyclomatic Complexity* dari *flowgraph* diatas.

(3) Jalur 1 = 1–2–3–4–5 merupakan jalur *independen flowgraph*.

***Tabel* 4.24** *Grafik matriks* Halaman Status

	1	**2**	**3**	**4**	**5**	E-1
1	0	1	0	0	0	1-1=0
2	0	0	1	0	0	1-1=0
3	0	0	0	1	0	1-1=0
4	0	0	0	0	1	1-1=0
5	0	0	0	0	0	0
	SUM (E + 1)					0+1=1

BAB V

PENUTUP

A. Kesimpulan

Berdasarkan hasil penelitian dan menjawab rumusan masalah dalam skripsi berjudul "Aplikasi Berbasis Web untuk Penanganan Pelaporan Kehilangan Bagasi di Bandara Nop Goliat Dekai", dapat disimpulkan bahwa aplikasi yang dikembangkan berhasil dirancang secara efektif untuk menangani proses pelaporan kehilangan bagasi secara digital. Aplikasi ini menyediakan fitur pelaporan online, verifikasi otomatis data laporan, pelacakan real-time terhadap bagasi yang hilang, serta pengiriman notifikasi status kepada penumpang. Inovasi ini meningkatkan efisiensi operasional karena mengurangi ketergantungan pada proses manual yang selama ini dinilai lambat dan rawan kesalahan. Selain itu, dengan adanya transparansi informasi melalui notifikasi yang dikirim secara berkala, penumpang merasa lebih tenang dan percaya terhadap proses pencarian bagasi. Pengujian sistem yang dilakukan melalui metode Black Box dan White Box menunjukkan bahwa aplikasi berjalan dengan baik sesuai fungsinya. Secara keseluruhan, aplikasi ini mampu meningkatkan pelayanan publik di bandara dengan memanfaatkan teknologi digital yang tepat guna dan dapat menjadi model solusi serupa di sektor transportasi udara lainnya.

B. Saran

Berdasarkan hasil penelitian dan implementasi aplikasi berbasis *web* untuk penanganan pelaporan kehilangan bagasi di Bandara NOP Goliat Dekai, terdapat

beberapa saran yang dapat dijadikan acuan untuk pengembangan lebih lanjut. Pertama, pengembangan aplikasi dapat ditingkatkan dengan menambahkan fitur-fitur tambahan seperti pelaporan berbasis suara, integrasi pelacakan bagasi melalui GPS, dan chatbot interaktif guna meningkatkan kenyamanan dan kecepatan pelayanan. Kedua, untuk menjamin kelancaran operasional, aplikasi ini sebaiknya diintegrasikan dengan sistem manajemen bandara yang lebih luas, seperti sistem check-in, data penerbangan, dan sistem keamanan bagasi, agar proses verifikasi dan pelacakan dapat dilakukan secara otomatis dan akurat. Ketiga, aspek keamanan data perlu mendapat perhatian serius dengan penerapan *enkripsi*, autentikasi dua faktor (2FA), dan pencatatan aktivitas pengguna (*audit log*) guna melindungi data pribadi dan informasi penting dari potensi penyalahgunaan. Keempat, disarankan kepada pihak bandara untuk melakukan pelatihan dan sosialisasi penggunaan aplikasi secara rutin kepada staf dan penumpang agar mereka memahami alur pelaporan dan dapat memanfaatkan sistem secara maksimal. Terakhir, evaluasi berkala terhadap performa aplikasi serta pengumpulan umpan balik dari pengguna sangat penting dilakukan guna memastikan aplikasi terus berkembang sesuai kebutuhan dan harapan pengguna di masa mendatang.

DAFTAR PUSTAKA

Sherina Kumala Dewi (2022). Analisis Penanganan Kehilangan Bagasi Penumpang dan Bagasi Tidak Bertuan Garuda Indonesia oleh *Gound Handling* PT Gapura Angkasa di Bandar Udara Juanda Surabaya. *Religional Education Social Laa Roiba Journal*. 4(5), 2-4.

Djoko Harsono, Imam Maliki. (2021). Perancangan Aplikasi Pelaporan Barang Hilang di KRL Commuterline Berbasis Web" *Jurnal Informatika & Komputasi*, 15(1), 82-87.

Inneke Febryana Disastra, Gallis Nawang Ginusti (2023). Pengaruh Penanganan Bagasi Terhadap Kepuasan Penumpang Maskapi Citilink Oleh Gapura Angkasa di Bandar Udara Internasional Juanda. *Jurnal Kewarganegaraan*. 6(1) 3-4.

Fitri, r., kom, s., & kom, m. (2020). Pemrograman basis data menggunakan mysql. Deepublish.

IM Adnyana, Suan Nauli, (2018), Penggunaan Sistem Aplikasi Out Of Gauge Dengan Baggage Handling System pada Terminal 3 Ultimate Domestik di Bandara Soekarno- Hatta. *Jurnal Manajemen Bisnis Transportasi & Logistik.*" 4(3) 1-16.

Krisbiantoro, d., kom, m., abda'u, p. D., & kom, m. (2021). *Dasar pemrograman web dengan bahasa html, php, dan database mysql* (vol. 1). Zahira media publisher.

Limbong, t. (2021). Pemrograman web dasar.

Nugroho, Y. S., & Ifada, N. (2021). Rancang Bangun Sistem Informasi Pelayanan Kehilangan Barang Berbasis Web. *Jurnal Informatika dan Rekayasa Perangkat Luna.,* 3(1), 53-60.

Oktaviani, N., Widiarta, I. M., & Nurlaily. (2019). Sistem Informasi Inventaris Barang Berbasis Web Pada Smp Negeri 1 Buer. *Jurnal Informatika, Teknologi Dan Sains*. *1*(2), 160–168. https://doi.org/10.51401/jinteks.v1i2.422

Rahimudin, Widiya afni Aprilia. (2023). Peran Unit Lost And Found Dalam Penanganan Bagasi Bermasalah Pada Maskapai Citilink Di Pt Gapura Angkasa Bandar Udara Internasionalzainuddin Abdul Madjid Lombok. *Populer: Jurnal Penelitian Mahasiswa.* 2(2), 44-55.

Rahmawati, E., & Susanto, H. (2022). Pengembangan Sistem Informasi Penanganan Bagasi Berbasis Web di Bandara Internasional Yogyakarta". *Jurnal Teknologi Informasi dan Ilmu Komputer.* 9(4), 321-330.

Sadiku, M. N., Ashaolu, T. J., & Musa, S. M. (2019). Smart airports. *International Journal of Scientific Advances.* 1(1), 27-29.