

Perancangan Aplikasi LAPIM (Laporan Insiden Masyarakat) Pada Dinas Perhubungan Pusat Data Dan Informasi DKI Jakarta Berbasis Website

Ryandaru Aryasatya¹, Noer Hikmah², Rangga Pebrianto³

^{1,2,3}Universitas Bina Sarana Informatika

Jl. Kramat Raya No 98 - Jakarta 10450, Indonesia

e-mail: ¹ryandaru.satya@gmail.com, ²noer.nhh@bsi.ac.id, ³rangga.rpo@bsi.ac.id

Artikel Info : Diterima : 25-06-2024 | Direvisi : 30-09-0000 | Disetujui : 01-10-2024

Abstrak - Aplikasi LAPIM (Laporan Insiden Masyarakat) merupakan solusi inovatif yang dikembangkan oleh Dinas Perhubungan Pusat Data dan Informasi DKI Jakarta untuk meningkatkan efisiensi dalam penanganan insiden transportasi. LAPIM memungkinkan masyarakat Jakarta melaporkan secara langsung insiden-insiden seperti parkir liar, masalah transportasi umum, dan kondisi jalanan berbahaya melalui platform berbasis website. Saat ini, terdapat beberapa permasalahan utama yang perlu diatasi, yaitu tingkat keterbatasan informasi mengenai laporan insiden terkait parkir liar, masalah transportasi umum, dan jalanan berbahaya di Jakarta, serta keterlambatan dalam pelaporan insiden dari masyarakat. Selain itu, tantangan lain adalah bagaimana membuat aplikasi LAPIM yang dapat mempermudah karyawan Dishub dalam menjalankan tugas mereka. Tujuan utama aplikasi ini adalah untuk memfasilitasi pelaporan insiden secara cepat dan akurat, serta memberikan data yang terpercaya kepada pihak berwenang untuk respons yang lebih efektif. Dengan fitur-fitur notifikasi real-time dan sistem pelacakan status laporan, LAPIM mempermudah interaksi antara masyarakat dan pihak terkait, meningkatkan responsibilitas dan transparansi dalam penanganan masalah transportasi. Integrasi teknologi dalam LAPIM tidak hanya mengoptimalkan pengumpulan dan manajemen data, tetapi juga menguatkan partisipasi publik dalam memantau dan meningkatkan kondisi transportasi kota. Pengembangan LAPIM sebagai platform digital menegaskan komitmen DKI Jakarta untuk menjadi *smart city* yang responsif terhadap kebutuhan masyarakat dalam mengelola infrastruktur transportasi. Diharapkan, keberhasilan LAPIM sebagai model aplikasi pelaporan insiden transportasi dapat menjadi inspirasi bagi pengembangan teknologi serupa di kota-kota besar lainnya, menjadikan transportasi lebih aman, efisien, dan berkelanjutan bagi seluruh penduduk Jakarta.

Kata Kunci : LAPIM, Laporan Insiden Masyarakat, Transportasi, Efisiensi

Abstracts - The LAPIM (Community Incident Report) application is an innovative solution developed by the Central Data and Information Department of the Jakarta Transportation Agency to enhance the efficiency of handling transportation incidents. LAPIM allows Jakarta residents to directly report incidents such as illegal parking, public transportation issues, and hazardous road conditions through a web-based platform. Currently, several main problems need to be addressed, including the limited information regarding incident reports related to illegal parking, public transportation issues, and hazardous road conditions in Jakarta, as well as delays in incident reporting from the public. Additionally, there is a challenge in creating an application that can facilitate the work of Transportation Agency employees. The primary goal of this application is to facilitate quick and accurate incident reporting and provide reliable data to authorities for more effective responses. With real-time notification features and report status tracking systems, LAPIM simplifies interactions between the public and relevant parties, enhancing accountability and transparency in handling transportation issues. The integration of technology in LAPIM not only optimizes data collection and management but also strengthens public participation in monitoring and improving the city's transportation conditions. The development of LAPIM as a digital platform underscores Jakarta's commitment to becoming a smart city responsive to the needs of its residents in managing transportation infrastructure. It is hoped that the success of LAPIM as a transportation incident reporting application model can inspire the development of similar technologies in other major cities, making transportation safer, more efficient, and sustainable for all Jakarta residents.

Keywords : LAPIM, Community Incident Reporting, Transportation, Efficiency



PENDAHULUAN

Pada era komputer dan internet saat ini, penggunaan teknologi informasi dalam layanan publik menjadi sangat penting untuk meningkatkan efisiensi dan efektivitas layanan yang diberikan kepada masyarakat. Dinas Perhubungan (Dishub) Pemerintah DKI Jakarta terus mengintegrasikan teknologi ke semua layanan untuk memenuhi kebutuhan masyarakat dengan cepat dan tepat (Handayani *et al.*, 2021). Salah satu upaya terbaru adalah pembuatan aplikasi yang disebut "LAPIM" (Laporan Insiden Masyarakat).

Aplikasi ini dirancang sebagai *platform* bagi warga untuk menyampaikan segala bentuk laporan dan keluhan yang berkaitan dengan infrastruktur perhubungan, pelayanan, hingga insiden yang terjadi di wilayah DKI Jakarta. Melalui aplikasi LAPIM, Dishub DKI Jakarta ingin memastikan bahwa setiap masukan dari masyarakat dapat diterima, dianalisis, dan ditindaklanjuti dengan cara yang sistematis dan terorganisir (Syafei *et al.*, 2019).

Pengembangan aplikasi ini memungkinkan interaksi langsung antara masyarakat dan Dishub secara *real-time*. Aplikasi LAPIM diharapkan tidak hanya memudahkan warga dalam melaporkan masalah tetapi juga mempercepat Dishub untuk menemukan dan menyelesaikan masalah. Selain berfungsi sebagai aplikasi pelaporan insiden, LAPIM juga merupakan *platform* yang melacak lokasi petugas yang sedang menangani masalah di wilayah Jakarta secara *real-time*.

Dishub DKI Jakarta berharap dapat memberikan layanan yang lebih baik kepada masyarakat dan meningkatkan respons terhadap berbagai insiden jalanan dengan menggunakan teknologi informasi dan komunikasi (Resty, 2024). Aplikasi LAPIM diharapkan dapat meningkatkan kenyamanan, keamanan, dan keselamatan pengguna transportasi umum di Jakarta. Aplikasi ini memungkinkan pengguna melaporkan insiden dengan cepat dan mudah.

Konsep dasar sistem adalah komponen yang saling terkait dan bekerja sama untuk mencapai tujuan tertentu melalui proses yang terorganisir (Triansyah, 2024). Dalam aplikasi LAPIM, sistem terdiri dari *input*, *proses*, *output* dan umpan balik. Sistem ini dirancang untuk memfasilitasi pencatatan, pengelolaan, dan analisis dari insiden-insiden yang dilaporkan oleh masyarakat.

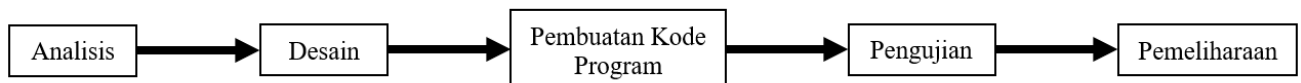
Aplikasi LAPIM mempunyai sistem berbasis web yang dapat mengimplementasikan pendekatan berorientasi objek. Model laporan berbasis web adalah sistem yang memungkinkan pengguna untuk membuat, mengelola, dan mendistribusikan laporan melalui internet (Sintaro, 2022). Laporan-laporan tersebut terdiri atas laporan insiden masyarakat, parkir liar, dan sebagainya.

Teknologi informasi yang terintegrasi dalam sistem LAPIM meningkatkan efisiensi dan efektivitas pengelolaan insiden. Aplikasi berbasis *website* memungkinkan masyarakat mengakses dan melaporkan insiden kapan saja dan dimana saja, serta memungkinkan Dinas perhubungan untuk merespons insiden secara sistematis dengan cepat dan tepat.

METODE PENELITIAN

Metode yang digunakan untuk membuat aplikasi LAPIM berbasis *website* adalah metode *waterfall*. Metode *waterfall* adalah model pengembangan perangkat lunak yang paling umum digunakan dalam pembuatan suatu sistem. Pengembangan ini mengikuti urutan linier dari perencanaan hingga pemeliharaan, di mana setiap tahap harus diselesaikan sebelum langkah ke tahap berikutnya, serta tidak ada kemungkinan untuk mengulang ke tahap sebelumnya (Widya Ningsih, 2023).

Adapun tahapan pada metode pengembangan *waterfall* yang akan digunakan sebagai berikut:



Sumber: Widya Ningsih (2023)

Gambar 1. Metode *Waterfall*

1. Analisis Kebutuhan Perangkat Lunak
Analisis bertujuan untuk mengumpulkan informasi tentang perangkat lunak yang digunakan. Pengguna yang terlibat ada 3 kategori yaitu, admin, petugas, dan *user*.
2. Desain
Tahap ini terfokus pada rancangan antarmuka, struktur basis data dan rancangan struktur navigasi. Rancangan antarmuka meliputi tampilan untuk admin, petugas, dan *user*.
3. Pembuatan Kode Program
Setelah desain dibuat, dilakukan pengkodean pada program yang dirancang menggunakan *ReactJS*, *Tailwind CSS*, *JSON*, *Node* dan *software Visual Studio Code* untuk menulis program
4. Pengujian

Langkah selanjutnya untuk memastikan program berjalan dengan sesuai, dilakukan pengujian dengan menggunakan metode blackbox testing untuk menguji hak akses admin, hak akses petugas, hak akses *user*.

5. Pemeliharaan (*Maintenance*)

Pemeliharaan mencakup koreksi kesalahan yang mungkin tidak terdeteksi pada tahap sebelumnya, melaksanakan perbaikan atas implementasi program unit, serta pengembangan layanan sistem dan penambahan persyaratan baru.

HASIL DAN PEMBAHASAN

Bagian ini menyajikan hasil dari tahapan pengembangan perangkat lunak yang telah dilakukan, serta menganalisis dan mendiskusikan temuan-temuan utama. Proses ini melibatkan beberapa langkah krusial, mulai dari analisis kebutuhan perangkat lunak hingga pemeliharaan sistem yang dikembangkan. Setiap langkah dibahas secara rinci untuk memberikan gambaran yang jelas tentang metodologi yang diterapkan dan bagaimana hal tersebut berkontribusi terhadap tercapainya tujuan penelitian.

Langkah pertama adalah analisis kebutuhan perangkat lunak, yang diikuti oleh desain sistem. Selanjutnya, dilakukan penyusunan *Logical Record Structure* (LRS) dan pembuatan *Entity Relationship Diagram* (ERD). Setelah itu, aktivitas pengembangan meliputi pembuatan *Activity Diagram*, pembuatan kode program, pengujian perangkat lunak, dan terakhir, pemeliharaan sistem.

1. Analisis Kebutuhan Perangkat Lunak

Analisis yang dilakukan meliputi:

a. Deskripsi proyek

Aplikasi LAPIM mempunyai tujuan untuk mempermudah masyarakat dalam melaporkan insiden kejadian yang sedang dialami oleh masyarakat.

b. Kebutuhan pengguna

1) Admin

- a) Admin memiliki kemampuan untuk mengelola pengguna aplikasi.
- b) Admin dapat melihat dan mengelola laporan insiden yang masuk melalui aplikasi.
- c) Admin bertanggung jawab untuk memastikan keamanan dan keandalan aplikasi.

2) Petugas

- a) Petugas dapat mengakses laporan insiden yang masuk ke dalam sistem.
- b) Petugas mempunyai kemampuan untuk menugaskan diri atau timnya untuk menangani insiden tertentu.
- c) Petugas perlu memperbarui status insiden di dalam sistemnya, menandai laporan sebagai “selesai” setelah menangani insiden.
- d) Petugas dapat berkomunikasi kembali dengan pelapor untuk mendapatkan informasi tambahan terkait dengan insiden yang dilaporkan.

3) *User*

- a) *User* perlu mendaftarkan akun di aplikasi LAPIM agar dapat melakukan pelaporan.
- b) *User* dapat melakukan pelaporan insiden pada fitur laporan yang mengharuskan pengguna untuk mengisi detail insiden seperti *username*, kategori, tanggal, lokasi, deskripsi dan foto untuk bukti insiden.
- c) *User* dapat memantau status laporan setelah dikirim.

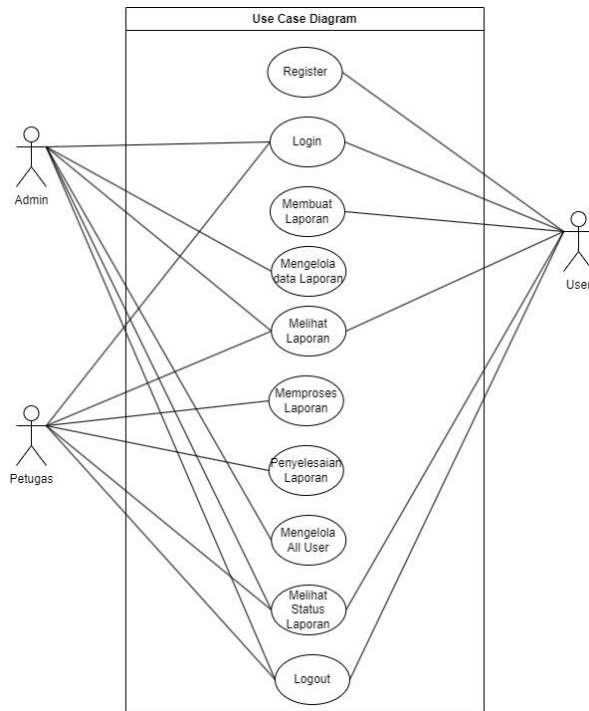
c. Analisa Masalah

Ada beberapa masalah yang ada dalam masyarakat tentang melaporkan insiden, serta bagaimana aplikasi LAPIM dapat membantu dalam menyelesaikannya, beberapa masalah tersebut antara lain:

- 1) Kesulitan dalam melaporkan insiden
- 2) Keterbatasan akses informasi
- 3) Kurangnya respons dalam melaporkan insiden
- 4) Pelacakan dan dokumentasi yang lemah

d. *Use Case Diagram*

Use case diagram adalah jenis interaksi yang terjadi antara pengguna dan sistem. *Use case* digunakan untuk menentukan perilaku sistem yang akan dibangun. Sebuah *use case* menunjukkan bagaimana pengguna akan berinteraksi dengan sistem yang ada (Ihramsyah, 2023).



Sumber: Hasil Penelitian (2024)

Gambar 2. Usecase Diagram

2. Desain

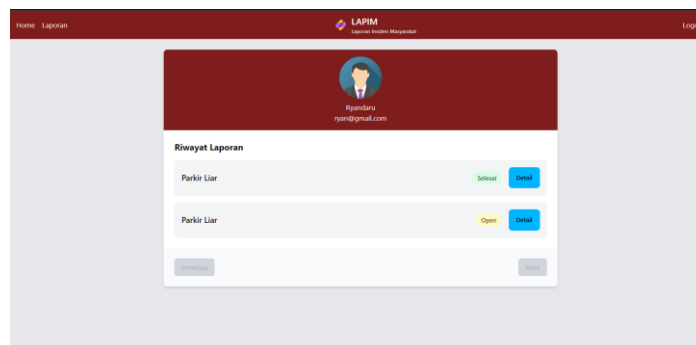
Berikut adalah beberapa tampilan desain dari aplikasi LAPIM:



Sumber: Hasil Penelitian (2024)

Gambar 3. Halaman *Homepage* LAPIM

Gambar tersebut adalah halaman homepage pada aplikasi LAPIM. Pada halaman tersebut terdapat tombol *home*, *about*, riwayat laporan terakhir, *login*, dan *register*.

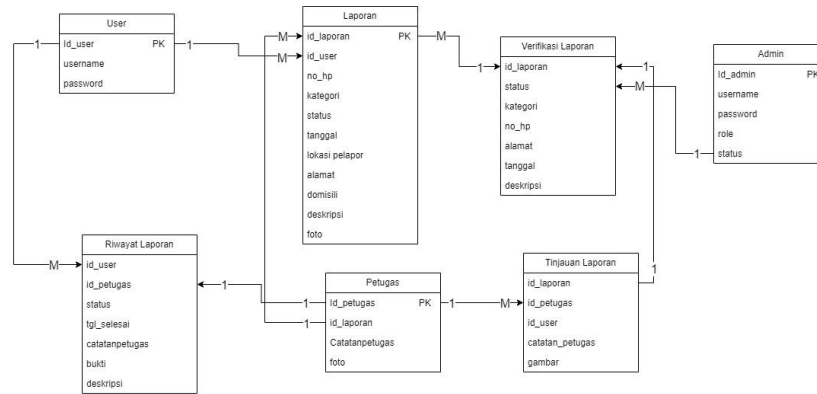


Sumber: Hasil Penelitian (2024)

Gambar 4. Halaman Riwayat Laporan *User*

Gambar tersebut merupakan halaman riwayat laporan bagian *user*. Bagian melihat riwayat laporan berfungsi untuk menyajikan daftar laporan yang dibuat oleh pengguna dalam periode tertentu, bagian ini memungkinkan pengguna untuk melihat laporan yang telah dibuat sebelumnya, serta meninjau status dan detail laporan insiden.

3. LRS (Logical Record Structure)



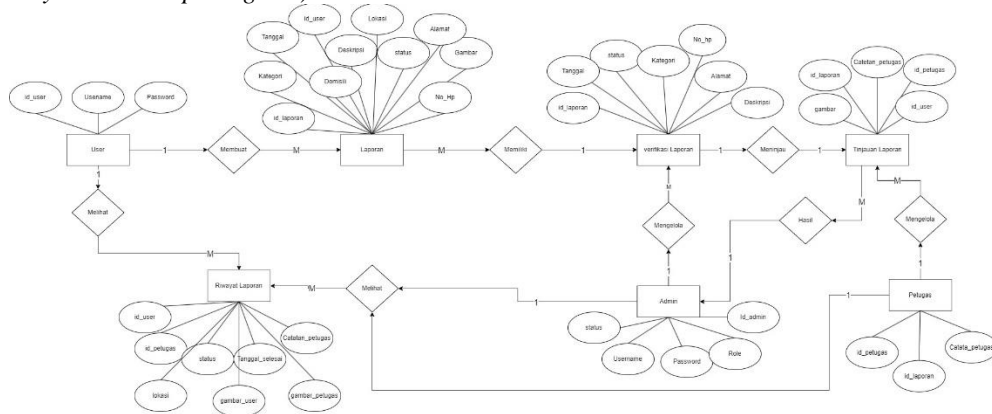
Sumber: Hasil Penelitian (2024)

Gambar 5. Logical Record Structure LAPIM

Logical Record Structure (LRS) adalah gambaran dari susunan rekaman berdasarkan tabel yang terbentuk dari hasil kombinasi berbagai himpunan entitas (Oktaviani *et al.*, 2019).

LRS di atas adalah struktur data yang dipakai untuk mengatur dan menyimpan informasi secara logis dalam sebuah sistem basis data. Pada perancangan aplikasi LAPIM, LRS berperan penting dalam pengorganisasian dan pengelolaan data laporan insiden yang diterima dari masyarakat.

4. ERD (Entity Relationship Diagram)



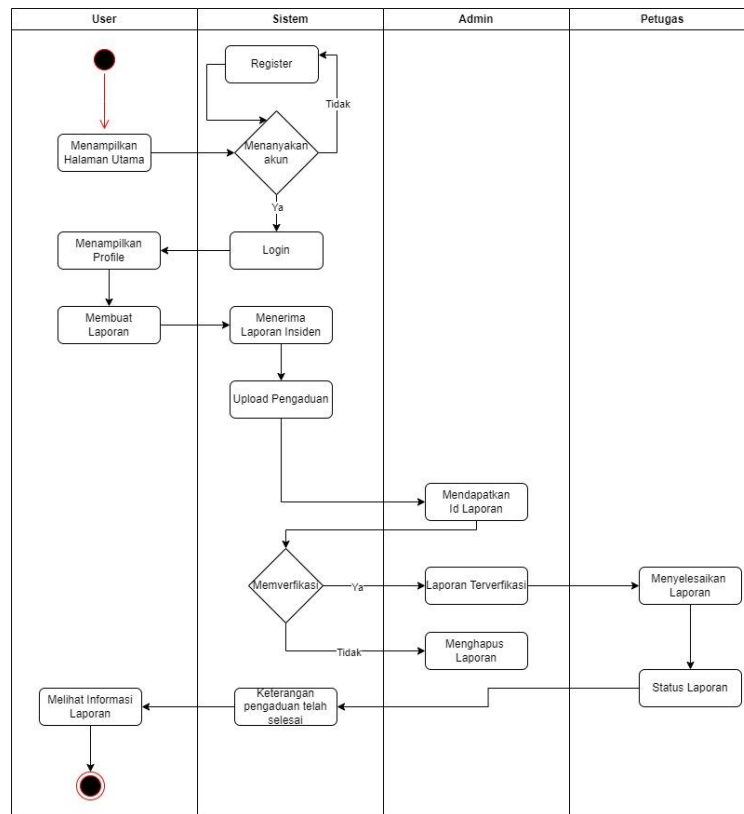
Sumber: Hasil Penelitian (2024)

Gambar 6. Entity Relationship Diagram LAPIM

ERD berfungsi untuk menggambarkan hubungan antara entitas penyimpanan data dalam diagram alir menggunakan berbagai notasi atau simbol untuk memvisualisasikan dan menggambarkan hubungan antar data tersebut (Emster *et al.*, 2021).

Diagram ERD di atas menggambarkan hubungan antara entitas-entitas utama dalam sistem pelaporan insiden masyarakat. Entitas yang terlibat mencakup *user*, registrasi, *login*, data laporan, petugas, dan penyelesaian laporan. ERD ini memvisualisasikan bagaimana data terkait laporan insiden dikumpulkan, dikelola, dan diakses oleh berbagai entitas-entitas tersebut.

5. Activity Diagram



Sumber: Hasil Penelitian (2024)

Gambar 7. Activity Diagram

Activity diagram di atas menggambarkan alur kerja yang terjadi dalam sistem pelaporan insiden masyarakat. Diagram ini menunjukkan langkah-langkah yang diambil oleh user, sistem, admin, dan petugas dari pembuatan laporan, verifikasi, penugasan petugas, hingga penyelesaian insiden. Setiap aktivitas yang terjadi sepanjang proses pelaporan diilustrasikan dengan jelas, membantu dalam memahami dan mengoptimalkan alur kerja untuk respons yang cepat dan efektif terhadap insiden.

6. Pembuatan Kode Program

Berikut ini adalah bagian dari pembuatan kode program aplikasi LAPIM yang menggunakan bahasa pemrograman Javascript (Node express) untuk pengembangan sisi server (backend). dan JSON untuk pengembangan sisi database.

```

23 fetch('http://localhost:3000/laporan')
24 .then(response => response.json())
25 .then(data => {
26   const openReports = data.filter(report => report.status === 'Terverifikasi');
27   const ids = openReports.map(report => report.idLaporan);
28   setReportIDs(ids);
29   setIsLoading(false);
30 });
31 .catch(error => {
32   console.error('Error fetching report IDs:', error);
33   setIsLoading(false);
34 });
35 }, []);
36
37 useEffect(() => {
38   if (selectedReportID) {
39     console.log('Fetching details for report ID:', selectedReportID); // Debug log
40     fetch(`http://localhost:3000/laporan/${selectedReportID}`)
41       .then(response => response.json())
42       .then(data => {
43         console.log('Fetched report details:', data); // Debug log
44         setReportDetails(data);
45         setShowReportDetailsModal(true);
46       })
47       .catch(error => {
48         console.error('Error fetching report details:', error);
    
```

Sumber: Hasil Penelitian (2024)

Gambar 8. Pembuatan Kode Program

7. Pengujian

Halaman pengujian menekankan pada pengujian aplikasi sebelum dirilis ke pengguna. Berikut adalah tahapan pengujian yang dilakukan menggunakan metode *blackbox*:

1. Rencana pengujian

a. Pengujian fungsional

Jenis pengujian yang memastikan bahwa setiap fitur atau fungsi dari sistem berjalan sesuai dengan spesifikasi yang diharapkan. Proses pengujian melibatkan pengujian kasus penggunaan, pengujian skenario, dan verifikasi *input-output*.

b. Pengujian non fungsional

Pengujian non-fungsional menunjukkan seberapa baik suatu perangkat lunak terhadap proses pengembangan, standar, dan waktu. Pengujian ini tidak menunjukkan layanan spesifik sistem perangkat lunak (Sianturi *et al.*, 2021).

2. Hasil pengujian

Tabel 1. Hasil Pengujian *Login*

No	Skenario Pengujian	Test Case	Hasil yang diharapkan	Hasil pengujian	Kesimpulan
1	<i>Login</i> dengan kredensial yang valid	Masukkan email dan kata sandi yang benar lalu klik tombol <i>login</i>	Pengguna berhasil masuk ke aplikasi dan diarahkan ke halaman utama	Pengguna berhasil masuk dan diarahkan ke halaman utama	Berhasil
2	<i>Login</i> dengan email yang salah	Masukkan email yang salah dan kata sandi yang benar lalu klik tombol <i>login</i>	Muncul pesan error "Email atau kata sandi salah"	Muncul pesan error "Email atau kata sandi salah"	Berhasil
3	<i>Login</i> dengan email tidak terdaftar	Masukkan email yang tidak terdaftar dan kata sandi yang benar lalu klik tombol <i>login</i>	Muncul pesan error "Email tidak terdaftar"	Muncul pesan error "Email tidak terdaftar"	Berhasil

Sumber: Hasil Penelitian (2024)

Tabel 2. Hasil Pengujian *Register*

No	Skenario Pengujian	Test Case	Hasil yang diharapkan	Hasil pengujian	Kesimpulan
1	Pendaftaran dengan data lengkap	Buka aplikasi, Pilih opsi pendaftaran, Isi semua kolom (nama, email, kata sandi, konfirmasi kata sandi) lalu klik daftar	Akun berhasil dibuat dan muncul notifikasi konfirmasi pendaftaran	Akun berhasil dibuat dan muncul notifikasi konfirmasi pendaftaran	Berhasil
2	Pendaftaran dengan email sudah terdaftar	Isi semua kolom dengan email yang sudah terdaftar lalu klik daftar	Muncul pesan error "Email sudah terdaftar"	Muncul pesan error "Email sudah terdaftar"	Berhasil
3	Pendaftaran dengan kolom kosong	Biarkan beberapa kolom wajib kosong lalu klik daftar	Muncul pesan error "Kolom wajib tidak boleh kosong"	Muncul pesan error "Kolom wajib tidak boleh kosong"	Berhasil

Sumber: Hasil Penelitian (2024)

Tabel 3. Hasil Pengujian Pembuatan Laporan

No	Skenario Pengujian	Test Case	Hasil yang diharapkan	Hasil pengujian	Kesimpulan
1	Membuat laporan dengan data lengkap	Isi semua kolom (jenis insiden, lokasi, deskripsi, foto) lalu klik kirim	Laporan berhasil dikirim dan notifikasi	Laporan berhasil dikirim dan notifikasi	Berhasil

2	Membuat laporan dengan kolom kosong	Biarkan kolom wajib kosong (misalnya jenis insiden atau lokasi) lalu klik kirim	konfirmasi pengiriman muncul Muncul pesan error "Kolom wajib tidak boleh kosong"	konfirmasi pengiriman muncul Muncul pesan error "Kolom wajib tidak boleh kosong"	Berhasil
3	Membuat laporan tanpa foto	Isi semua kolom kecuali foto lalu klik kirim	Muncul pesan error "wajib mengupload foto bukti"	Muncul pesan error "wajib mengupload foto bukti"	Berhasil

Sumber: Hasil Penelitian (2024)

Tabel 4. Hasil Pengujian Verifikasi Laporan

No	Skenario Pengujian	Test Case	Hasil yang diharapkan	Hasil pengujian	Kesimpulan
1	Admin memeriksa detail laporan	Admin membuka laporan dan melihat detail insiden yang dilaporkan.	Detail laporan ditampilkan dengan jelas, termasuk foto, lokasi, dan deskripsi insiden.	Sesuai dengan harapan	Valid
2	Admin menandai laporan sebagai terverifikasi	Admin menandai laporan sebagai terverifikasi setelah memverifikasi informasi.	Laporan ditandai sebagai terverifikasi dan siap untuk diproses lebih lanjut.	Sesuai dengan harapan	Valid
3	Admin menugaskan laporan kepada petugas lapangan	Admin menugaskan laporan kepada petugas lapangan untuk ditangani.	Laporan dengan status "terverifikasi" dapat di proses oleh petugas	Sesuai dengan harapan	Valid

Sumber: Hasil Penelitian (2024)

Tabel 5. Hasil Pengujian Peninjauan Laporan

No	Skenario Pengujian	Test Case	Hasil yang diharapkan	Hasil pengujian	Kesimpulan
1	Petugas meninjau detail laporan	Petugas membuka laporan dan melihat detail insiden yang dilaporkan.	Detail laporan ditampilkan dengan jelas, termasuk foto, lokasi, dan deskripsi insiden.	Sesuai dengan harapan	Valid
2	Petugas menugaskan laporan untuk tindakan lebih lanjut	Petugas menugaskan laporan kepada tim lapangan untuk ditangani.	Laporan berhasil ditugaskan dan tim lapangan menerima tugas laporan.	Sesuai dengan harpaan	Valid
3	Petugas memberikan tanggapan kepada pelapor	Petugas memberikan tanggapan atau pembaruan status kepada pelapor	Pelapor menerima tanggapan atau pembaruan status	Sesuai dengan harapan	Valid
4	Petugas mengunggah bukti Tindakan dan menambahkan catatan	Petugas mengunggah foto atau dokumen sebagai bukti tindakan yang telah dilakukan.	Bukti tindakan berhasil diunggah dan dilampirkan pada laporan.	Sesuai dengan harapan	Valid
5	Petugas menyelesaikan laporan	Petugas menandai laporan sebagai selesai setelah tindakan diambil.	Laporan ditandai sebagai selesai dan pelapor menerima insiden telah ditangani.	Sesuai dengan harapan	Valid

Sumber: Hasil Penelitian (2024)

Tabel 6. Hasil Pengujian Melihat Riwayat Laporan

No	Skenario Pengujian	Test Case	Hasil yang diharapkan	Hasil pengujian	Kesimpulan
1	Pengguna membuka riwayat laporan	Pengguna masuk ke aplikasi dan membuka menu riwayat laporan.	Daftar semua laporan yang telah dikirim oleh pengguna ditampilkan dengan ringkasan singkat (misalnya, tanggal, status, dan judul laporan).	Sesuai dengan harapan	Valid
2	Pengguna melihat detail laporan tertentu	Pengguna memilih laporan tertentu dari daftar riwayat.	Detail lengkap laporan ditampilkan, termasuk deskripsi insiden, lokasi, foto, status terkini, dan tanggapan dari Dishub.	Sesuai dengan harapan	Valid
3	Pengguna melihat status laporan terkini	Pengguna memeriksa status laporan yang sedang diproses.	Status terkini dari laporan ditampilkan dengan jelas (misalnya, "Diterima", "Sedang Diproses", "Selesai").	Sesuai dengan harapan	Valid

Sumber: Hasil Penelitian (2024)

8. Pemeliharaan (*Maintenance*)

Di bawah ini adalah spesifikasi *hardware* dan *software* yang dibutuhkan dalam aplikasi LAPIM:

1. Spesifikasi *Hardware*

a. *Server*:

- 1) CPU: Prosesor Intel Xeon E5-2680 v4 atau setara.
- 2) RAM: 64 GB DDR4.
- 3) Penyimpanan: SSD 1 TB.
- 4) Perangkat *Input*: *Mouse* dan *keyboard*.
- 5) Tampilan: Monitor dengan resolusi layar minimal 1024x768.
- 6) Koneksi internet: Minimal 2 Mbps.

b. *Client*:

- 1) CPU: Processor Pentium® 4.
- 2) RAM: DDR2 1 GB.
- 3) Penyimpanan: Hard Disk 20 GB.
- 4) Perangkat *Input*: *Mouse* dan *keyboard*.
- 5) Tampilan: Monitor dengan resolusi layar minimal 1024x768.
- 6) Koneksi internet: Minimal 56 kbps.

2. Spesifikasi *Software*

a. *Server*:

- 1) Sistem Operasi: *Microsoft Windows*.
- 2) *Framework*: LAPIM dikembangkan dengan *Node.js* dan *Express.js*.
- 3) *Browser*: LAPIM dapat diakses melalui berbagai jenis *web browser* termasuk *Google Chrome*.

b. *Client*:

- 1) Sistem Operasi: *Microsoft Windows*.
- 2) *Browser*: Aplikasi web yang direkomendasikan untuk mengakses LAPIM adalah *Google Chrome*.

KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan mengenai “perancangan aplikasi LAPIM (Laporan Insiden Masyarakat) pada Dinas Perhubungan Pusat Data dan Informasi DKI Jakarta berbasis *website*”, maka dapat diambil kesimpulan bahwa pembuatan aplikasi LAPIM sudah mencapai target yang diharapkan. Aplikasi ini berhasil mengatasi keterbatasan informasi dengan menyediakan platform bagi masyarakat untuk melaporkan insiden terkait parkir liar, masalah transportasi umum, dan jalanan berbahaya secara *real-time*. Hal ini meningkatkan ketersediaan dan akurasi informasi yang dapat diakses oleh Dishub. Selain itu, aplikasi LAPIM juga meningkatkan kesadaran dan ketepatan waktu dalam pelaporan insiden, memudahkan masyarakat untuk mengirim laporan dengan cepat dan mudah, sehingga partisipasi masyarakat dalam pelaporan insiden meningkat.

Berdasarkan hasil penelitian sebelumnya, saran atau rekomendasi dapat menjadi masukan bagi Dinas Perhubungan Pusat Data dan Informasi DKI Jakarta. Salah satu rekomendasi utama adalah untuk mendorong pengembangan aplikasi berdasarkan umpan balik pengguna. Hal ini melibatkan fitur-fitur yang lebih relevan dan memberikan nilai tambah yang signifikan sesuai dengan kebutuhan *user* yang selalu berubah.

REFERENSI

- Emster, M. Von, Abdullah, M. H., & Sabtu, J. (2021). Sistem Informasi Pengaduan Kekerasan Perempuan dan Anak pada DP3A Kota Ternate Berbasis *Website* dengan Menggunakan PHP dan Mysql. *JAMINFOKOM: Jurnal Ilmiah Manajemen Informatika & Komputer*, 1(1), 46–54. <https://jurnal.aikomternate.ac.id/index.php/jaminfokom>
- Handayani, S., Afrianti, D. A., Suryandari, M., & Bekasi, K. (2021). *Implementasi kebijakan angkutan umum di dki jakarta*. 2(1), 19–28.
- Ihramsyah, V. Y. (2023). *Perancangan Aplikasi Sistem Informasi Penjualan Makanan Cepat Saji Berbasis Web Studi Kasus Kedai Cheese.Box*. 4(April), 117–139.
- Oktaviani, A., Sarkawi, D., Priadi, A., Bina, U., Informatika, S., Bina, U., & Informatika, S. (2019). *Perancangan Sistem Parkir Pada Gedung Menara Palma Jakarta*. 12(2), 231–241.
- Resty, K. D. (2024). *Peran Dinas Perhubungan Kota Medan Dalam Pengawasan Pelaksanaan Standar Operasional Prosedur Bagi Pengemudi Angkutan Kota (Angkot) Di Kota Medan*. 4.
- Sianturi, R. A., Sinaga, A. M., Pratama, Y., Simatupang, H., Panjaitan, J., & Sihotang, S. (2021). Perancangan Pengujian Fungsional Dan Non Fungsional Aplikasi Siappara Di Kabupaten Humbang Hasundutan. *Jurnal Komputer Dan Informatika*, 9(2), 133–141. <https://doi.org/10.35508/jicon.v9i2.4706>
- Sintaro, S. (2022). *Permodelan Sistem Informasi Pembelian dan Penjualan Berbasis Website*. 1, 25–32.
- Syafei, I., Kamayani, M., & Sinduningrum, E. (2019). *Perancangan aplikasi pengaduan masyarakat terhadap lingkungan di tingkat kelurahan*. 4(2502), 111–116. <https://doi.org/10.22236/teknoka.v>
- Triansyah, F. A. (2024). *KONSEP DASAR SISTEM INFORMASI MANAJEMEN*. CV Rey Media Grafika.
- Widya Ningsih, H. N. (2023). *PERBANDINGAN MODEL WATERFALL DAN METODE PROTOTYPE UNTUK PENGEMBANGAN APLIKASI PADA SISTEM INFORMASI*. 5(1), 83–95.