

PERANCANGAN DAN PENGEMBANGAN APLIKASI WEB PELAPORAN KAMPUS DI UNIVERSITAS NEGERI MEDAN DENGAN INTEGRASI *CHATBOT*

Freyro Dobry Sianipar^{1*}, Muhammad Hidayatul Arifin², Windy Aulia³,
Muhammad Haikal Al Majid⁴, Adidtya Perdana⁵

Universitas Negeri Medan, Medan, Sumatera Utara, Indonesia¹

Email*: freyrodobry31@gmail.com

Universitas Negeri Medan, Medan, Sumatera Utara, Indonesia²

Email: arif.4233250005@mhs.unimed.ac.id

Universitas Negeri Medan, Medan, Sumatera Utara, Indonesia³

Email: windy.4231250021@mhs.unimed.ac.id

Universitas Negeri Medan, Medan, Sumatera Utara, Indonesia⁴

Email: muhammadhaikalalmajid.4232250004@mhs.unimed.ac.id

Universitas Negeri Medan, Medan, Sumatera Utara, Indonesia⁵

Email: adidtya@unimed.ac.id

(*) *Corresponding Author*

ABSTRAK

Sistem pelaporan konvensional di lingkungan kampus seringkali menghasilkan laporan yang tidak terdokumentasi dengan baik, yang menyebabkan proses penanganan menjadi lambat dan tidak jelas. Ini adalah masalah yang mendasari penelitian ini. Tujuan dari penelitian ini adalah untuk membuat dan mengembangkan aplikasi *web* pelaporan kampus yang terintegrasi dengan *chatbot* untuk meningkatkan transparansi, kecepatan, dan efisiensi dalam penyampaian dan penanganan laporan. Metode yang digunakan adalah *Research and Development* (R&D) dengan model pengembangan *Waterfall*, mencakup tahap analisis kebutuhan, perancangan sistem, implementasi, pengujian, dan evaluasi. Pengembangan aplikasi ini menggunakan bahasa pemrograman PHP untuk *backend*, MySQL sebagai *basis data*, dan *Tailwind CSS* dan *JavaScript* untuk antarmuka pengguna. *Chatbase* digunakan untuk mengintegrasikan *chatbot* untuk menjawab pertanyaan otomatis tentang proses pelaporan. Hasil penelitian menunjukkan bahwa aplikasi yang dikembangkan mampu memudahkan proses pelaporan secara sistematis melalui fitur formulir *input*, autentikasi pengguna, dan *dashboard* yang memantau *status* laporan, yang memudahkan pengelolaan laporan oleh pihak kampus. Selain itu, fitur *chatbot* membantu menjawab pertanyaan pengguna secara cepat dan tepat, meningkatkan interaksi dan kepuasan pengguna. Kesimpulan penelitian menunjukkan bahwa penggunaan aplikasi *web* pelaporan dengan *chatbot* dapat meningkatkan alur laporan, mempercepat respons penanganan, dan meningkatkan transparansi dan dokumentasi dalam pengelolaan laporan di lingkungan kampus.

Kata kunci: *Chatbot*, kampus, pelaporan, sistem informasi, *web*

ABSTRACT

Conventional reporting systems in the campus environment often produce reports that are not well documented, which causes the handling process to be slow and unclear. This is the problem underlying this research. The purpose of this research is to create and

develop a campus reporting web application integrated with a chatbot to increase transparency, speed, and efficiency in the delivery and handling of reports. The method used is Research and Development (R&D) with the Waterfall development model, including the stages of needs analysis, system design, implementation, testing, and evaluation. This application development uses PHP programming language for the backend, MySQL as the database, and Tailwind CSS and JavaScript for the user interface. Chatbase is used to integrate a chatbot to answer automated questions about the reporting process. The results showed that the developed application was able to facilitate the reporting process systematically through the input form feature, user authentication, and a dashboard that monitors the status of the report, which facilitates the management of reports by the campus. In addition, the chatbot feature helps answer user questions quickly and precisely, increasing user interaction and satisfaction. The conclusion of the study shows that the use of a reporting web application with a chatbot can improve the flow of reports, accelerate handling responses, and increase transparency and documentation in report management in the campus environment.

Keywords: Chatbot, campus, reporting, information system, web

1. PENDAHULUAN

Universitas sebagai institusi pendidikan memiliki berbagai fasilitas dan lingkungan yang harus terjaga demi mendukung proses akademik dan kesejahteraan civitas akademika. Namun, dalam operasionalnya, sering kali terjadi berbagai permasalahan di lingkungan kampus, seperti fasilitas yang rusak, hingga kejadian yang sangat memerlukan perhatian pihak kampus seperti tindak pelanggaran aturan atau gangguan keamanan. Sayangnya, sistem pelaporan yang masih dilakukan secara konvensional sering kali kurang efektif dalam menangani masalah-masalah tersebut. Banyak laporan yang tidak terdokumentasi dengan baik, sehingga proses penyelesaian menjadi lambat dan tidak transparan [1]. Penggunaan teknologi informasi saat ini memungkinkan berbagai sistem berbasis digital untuk meningkatkan efisiensi dalam berbagai aspek, termasuk dalam pengelolaan laporan kejadian di kampus. Sistem pelaporan berbasis *web* bertujuan untuk dapat menjadi solusi yang memungkinkan mahasiswa dengan mudah menyampaikan laporan kejadian secara cepat dan terdokumentasi dengan baik. Dengan adanya sistem ini, setiap laporan dapat diproses lebih sistematis, pihak yang berwenang dapat segera mengambil tindakan, serta status penyelesaian laporan dapat dipantau dengan lebih transparan oleh pelapor [2].

Website telah menjadi sangat penting dalam era globalisasi karena perkembangan teknologi digital. *Website* tidak hanya berfungsi sebagai sarana utama untuk berinteraksi dan mengakses informasi, tetapi juga berfungsi sebagai alat untuk menyebarkan informasi, menjual barang, dan membentuk komunitas [3]. "*Website* sangat berguna sebagai gerbang utama menuju informasi dan interaksi", menurut sebuah penelitian yang dipublikasikan. Pengembangan aplikasi *web* memanfaatkan berbagai teknologi pendukung untuk memenuhi peran tersebut, seperti PHP dan SQL untuk mengelola data secara dinamis melalui *database* seperti MySQL, HTML untuk membangun struktur halaman, CSS untuk menyusun tampilan *visual*, dan JavaScript untuk menambahkan elemen interaktif. Dengan kombinasi teknologi-teknologi ini, situs *web* yang menarik, informatif, dan berguna dapat dibuat untuk memenuhi berbagai kebutuhan komunikasi dan transaksi di era digital [4].

Pemrograman *web* adalah proses membuat program atau aplikasi berbasis *web* dengan menggunakan bahasa pemrograman yang mengarahkan komputer untuk memproses data dan menampilkan informasi sesuai kebutuhan. Aplikasi *web* biasanya berfungsi untuk menampilkan informasi yang diminta pengguna. Pengguna menggunakan *browser* untuk mengirimkan permintaan ke *web server*. *Web server* memproses permintaan dan mengirimkan hasilnya kembali ke *browser* untuk dilihat. Aplikasi *web* terdiri dari tiga komponen berdasarkan mekanisme tersebut. Yang pertama menangani permintaan pengguna, yang kedua memodelkan masalah *domain* atau *business logic*, dan yang ketiga mengelola tampilan pengguna [5].

Penelitian sebelumnya yang dilakukan oleh Niatia dkk, menunjukkan penggunaan metode *Waterfall* dalam pengembangan sistem pengaduan berbasis *Android*. Tahapan ini, mulai dari analisis kebutuhan, desain sistem, penulisan kode, pengujian dengan metode *blackbox* dan *whitebox*, hingga penerapan dan pemeliharaan, dilakukan secara berurutan dan sistematis untuk memastikan kualitas sistem. Metode terstruktur ini tidak hanya membantu dalam dokumentasi menyeluruh proses pengembangan, tetapi juga meningkatkan kejelasan dan responsivitas dalam menangani keluhan dan pengaduan mahasiswa [6]. Metode ini dapat digunakan sebagai referensi untuk pengembangan sistem pelaporan kampus Universitas Negeri Medan.

Sebuah *chatbot*, yang juga disebut *smartbot* atau *chatterbot*, adalah sebuah program komputer yang memfasilitasi komunikasi antara manusia dan komputer, baik lewat *audio* maupun pesan tertulis. *Chatbot* dibuat sebagai sistem percakapan interaktif yang memiliki banyak tujuan, seperti memberikan layanan pelanggan, mengumpulkan informasi [7]. Dengan memanfaatkan teknologi kecerdasan buatan, *chatbot* bisa mengerti pertanyaan dari pengguna dan memberikan jawaban yang sesuai dan cepat, sehingga membantu dalam meningkatkan efisiensi serta personalisasi layanan di berbagai aplikasi, termasuk sistem pelaporan di perguruan tinggi.

Penelitian sebelumnya menggunakan pendekatan yang inovatif untuk memasukkan *chatbot* ke dalam *website* dengan tujuan meningkatkan interaksi dan efisiensi layanan. Penelitian tersebut melakukan ini dengan menyalin kode integrasi dari *platform WatsonX* dan menerapkannya langsung ke struktur kode *website* toko komputer. Metode ini menghasilkan tampilan *web* yang menampilkan *chatbot* yang terintegrasi, yang memungkinkan interaksi langsung dengan pengguna dan mempercepat penyebaran informasi. Dengan menggunakan metode ini, ada kemungkinan besar bahwa teknologi *chatbot* dapat digunakan untuk meningkatkan layanan seperti sistem pelaporan kampus, di mana responsivitas dan kecepatan sangat penting [7].

Dalam penelitian lain, *chatbot* digunakan pada *website* untuk meningkatkan layanan informasi secara *real-time*. Pada penelitian tersebut diterapkan pada *website* Universitas Amikom Purwokerto dengan mengembangkan *chatbot asisten virtual* dengan memasukkan teknologi ke dalam situs *web*nya. Sistem yang dibangun menggunakan *framework CodeIgniter* dan *basis data MySQL* dalam penelitian ini memungkinkan pengelolaan *database chat* dinamis dan *update* informasi yang terus menerus. *Chatbot* dapat berinteraksi secara bersamaan dan menjawab pertanyaan pengguna dengan cepat, meringankan tugas manajer layanan pelanggan dan mengurangi biaya operasional. Metode ini menunjukkan bahwa *chatbot* di *website* dapat meningkatkan produktivitas operasi dan memudahkan komunikasi dua arah antara pengguna dan *system* [8].

Berdasarkan uraian di atas, terdapat beberapa permasalahan yang perlu diselesaikan. Pertama, sistem pelaporan di kampus yang masih dilakukan secara konvensional menyebabkan laporan sering kali tidak terdokumentasi dengan baik,

sehingga proses penyelesaian menjadi lambat dan kurang transparan. Kedua, belum adanya integrasi teknologi *chatbot* dalam sistem pelaporan di Universitas Negeri Medan yang berpotensi meningkatkan efisiensi dalam penanganan laporan mahasiswa. Ketiga, diperlukan sebuah sistem berbasis *web* yang mampu memfasilitasi proses pelaporan secara cepat, terdokumentasi, dan dapat diakses dengan mudah oleh mahasiswa dan pihak berwenang. Oleh karena itu, penelitian ini bertujuan untuk merancang dan mengembangkan aplikasi *web* pelaporan kampus di Universitas Negeri Medan dengan integrasi *chatbot* guna meningkatkan efisiensi dan transparansi dalam penanganan laporan.

2. METODE

Penelitian ini menggunakan metode *Research and Development* (R&D) dengan model pengembangan perangkat lunak berbasis *Waterfall*, yang mencakup tahap analisis kebutuhan, perancangan sistem, implementasi dan pengujian. Teknologi yang digunakan dalam pengembangan aplikasi meliputi PHP sebagai bahasa pemrograman *backend*, MySQL sebagai *basis data*, serta *Tailwind* CSS dan JavaScript untuk tampilan antarmuka pengguna. Penelitian ini bertujuan untuk merancang dan mengembangkan Aplikasi *Web* Pelaporan Kejadian di Kampus yang memungkinkan mahasiswa untuk melaporkan kejadian dengan lebih cepat dan sistematis.

Analisis kebutuhan

Pada tahap ini dilakukan proses identifikasi kebutuhan pengguna terhadap sistem pelaporan yang akan dibangun. Metode yang digunakan dalam pengumpulan data adalah observasi dan diskusi informal dengan mahasiswa sebagai pengguna akhir. Dari proses ini diperoleh dua jenis kebutuhan, yaitu kebutuhan fungsional dan non-fungsional. Kebutuhan fungsional meliputi kemampuan sistem untuk menyediakan formulir pelaporan yang mencakup data Nama, NIM, Kelas, Fakultas, Program Studi, Kategori, dan Deskripsi. Sistem juga harus mampu menyimpan laporan ke dalam basis data dan memungkinkan pengguna untuk melihat status laporan berdasarkan NIM dan Nama. Sedangkan kebutuhan non-fungsional mencakup tampilan antarmuka yang responsif, kemudahan akses melalui *web browser*, serta kecepatan respon sistem yang optimal.

Perancangan sistem

Perancangan dilakukan untuk mengilustrasikan bagaimana sistem akan bekerja berdasarkan hasil analisis kebutuhan sebelumnya. Perancangan ini mencakup pembuatan flowchart untuk menggambarkan alur kerja sistem dan interaksi antara pengguna dengan sistem, di mana pengguna dapat mengirimkan laporan dan memeriksa status laporan. Selain itu, dilakukan perancangan basis data menggunakan struktur tabel yang disesuaikan dengan atribut pelaporan. Tabel utama yang digunakan memiliki *atribut* *id_laporan*, nama, NIM, kelas, fakultas, program studi, kategori, deskripsi, dan status laporan. Selanjutnya, perancangan antarmuka sistem dilakukan menggunakan HTML dan *Tailwind* CSS dengan memperhatikan aspek kemudahan penggunaan dan estetika tampilan. Beberapa halaman utama yang dirancang mencakup halaman utama, formulir pelaporan, status laporan, dan tampilan *chatbot*.

Database management

Database Management System merupakan *software* atau sistem yang dirancang untuk membantu dalam pemrosesan data dan menjalankan operasi yang diminta oleh

banyak pengguna *basis data*. Menjaga agar *basis data* selalu teratur dan mudah diakses merupakan tanggung jawab DBMS yang berfungsi sebagai penghubung antara *basis data* dan program aplikasi [9]. Maka, dapat disimpulkan bahwa DBMS akan mengatur segala akses ke *basis data* yang dilakukan oleh pengguna. Di penelitian ini menggunakan MySQL sebagai *database management*. MySQL merupakan sebuah sistem *basis data* sumber terbuka yang mampu melakukan pengolahan data secara bersamaan. Michael "Monty" Widenius mendirikan MySQL pada tahun 1995. Pada tahun 2000, MySQL diumumkan dengan lisensi ganda yang memungkinkan semua orang untuk menggunakannya secara gratis di bawah Lisensi Publik Umum GNU, yang kemudian membuatnya semakin populer. MySQL AB adalah perusahaan yang memiliki dan mengembangkan MySQL, yang kini menjadi anak perusahaan dari *Sun Microsystems*. MySQL begitu populer saat ini bukan tanpa alasan, tapi karena keunggulan fitur-fiturnya yang beragam, salah satunya adalah kecepatan yang menjadi ciri khasnya [10].

Implementasi dan pengujian

Implementasi merupakan tahap di mana aplikasi *web* pelaporan kampus yang telah dirancang mulai dikembangkan dan diuji untuk memastikan fungsionalitas serta keandalan sistem. Proses pengembangan dilakukan menggunakan teknologi berbasis *web* dengan menerapkan konsep pemrograman berbasis *server* dan *client-side*. Setiap komponen aplikasi, termasuk formulir pelaporan, sistem manajemen laporan, serta integrasi *chatbot*, dikembangkan secara bertahap sesuai dengan rancangan yang telah disusun pada tahap perancangan. Dalam pengembangannya, aplikasi ini dibangun menggunakan HTML, *Tailwind* CSS, JavaScript, dan PHP sebagai bahasa pemrograman utama. Sistem manajemen basis data menggunakan MySQL untuk menyimpan laporan yang diajukan oleh pengguna. Selain itu, fitur *chatbot* dirancang untuk memberikan respons otomatis terhadap pertanyaan yang sering diajukan oleh pengguna.

Setelah tahap pengembangan selesai, dilakukan pengujian sistem untuk memastikan bahwa aplikasi berjalan sesuai dengan spesifikasi yang telah ditetapkan. Pengujian dilakukan dengan metode *Black Box Testing*, yang berfokus pada pengujian fungsional tanpa melihat kode sumber. Pengujian ini mencakup validasi *input*, pengelolaan laporan, serta interaksi dengan *chatbot*. Hasil pengujian digunakan untuk memperbaiki kekurangan dan meningkatkan performa aplikasi sebelum diterapkan secara luas.

3. HASIL DAN PEMBAHASAN

Analisis kebutuhan

Tahap analisis kebutuhan dilakukan untuk mengidentifikasi fitur dan fungsi yang harus dimiliki oleh sistem pelaporan kampus berbasis *web*. Analisis ini didasarkan pada permasalahan yang sering dihadapi mahasiswa, seperti kesulitan menyampaikan laporan secara langsung, keterbatasan waktu, serta kurangnya informasi mengenai status laporan. Kebutuhan sistem diperoleh melalui diskusi informal dengan beberapa mahasiswa Universitas Negeri Medan yang pernah mengalami kejadian kampus namun kesulitan menyampaikannya ke pihak berwenang. Selain itu, peninjauan terhadap sistem pelaporan tradisional juga menunjukkan kurangnya efisiensi dan transparansi dalam proses pelaporan.

Berdasarkan hasil analisis tersebut, diperoleh kebutuhan sistem sebagai berikut: Mahasiswa harus dapat mengisi formulir pelaporan dengan atribut: Nama, NIM, Kelas, Fakultas, Program Studi, Kategori, dan Deskripsi, Sistem harus mampu menyimpan

laporan ke dalam basis data, Mahasiswa dapat melihat status laporan melalui halaman login menggunakan NIM dan Nama, Admin dapat melihat semua laporan yang masuk melalui panel administrator, Sistem menyediakan *chatbot* untuk memberikan informasi otomatis kepada pengguna terkait cara pelaporan, jenis laporan, dan panduan penggunaan sistem.

Perancangan sistem

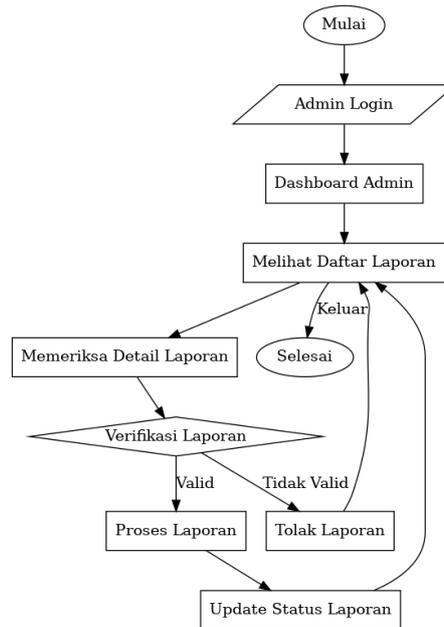
Hasil dari tahap analisis kebutuhan dituangkan ke dalam perancangan sistem pelaporan kampus berbasis web, dengan tujuan utama merancang alur kerja dan struktur sistem yang sesuai dengan kebutuhan pengguna. Perancangan dilakukan secara sistematis dengan memperhatikan aspek alur proses, desain *basis data*, dan antarmuka pengguna.

Dalam menggambarkan alur kerja sistem, digunakan *flowchart* untuk menjelaskan proses interaksi antara pengguna dan sistem. *Flowchart* dibagi menjadi dua, yaitu *flowchart* untuk pengguna (mahasiswa) dan admin: *Flowchart* Pengguna menggambarkan alur mulai dari *login* dengan NIM dan Nama, mengisi formulir pelaporan yang mencakup data diri dan deskripsi kejadian, kemudian sistem akan menyimpan data ke dalam *database*. Setelah laporan dikirim, pengguna dapat mengecek status laporan melalui menu yang tersedia. *Flowchart* Admin menunjukkan alur kerja admin dimulai dari proses *login*, melihat daftar laporan yang masuk, memverifikasi isi laporan, serta memperbarui status laporan agar dapat dilihat oleh mahasiswa.

Terakhir, dalam tahap perancangan ini juga disiapkan fitur integrasi *chatbot* yang akan membantu mahasiswa memperoleh jawaban otomatis terkait proses pelaporan dan status laporan mereka. Integrasi ini akan mempermudah komunikasi tanpa harus melalui admin secara langsung. Perancangan sistem ini menjadi fondasi penting dalam tahap implementasi, guna memastikan sistem berjalan sesuai skenario yang diharapkan dan memudahkan pengguna dalam melakukan pelaporan secara daring.

Flowchart proses pada admin

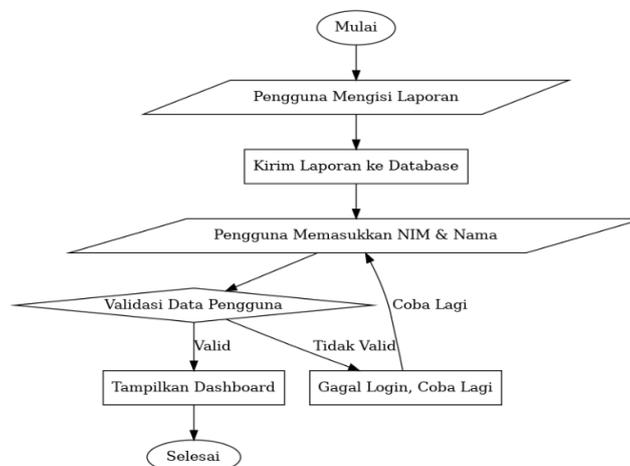
Pada Gambar 1 menjelaskan alur kerja sistem dalam menangani laporan yang masuk ke dalam aplikasi *web* pelaporan kampus. Alur ini dimulai ketika seorang pengguna mengirimkan laporan melalui formulir yang telah disediakan. Laporan tersebut kemudian tersimpan dalam basis data dan dapat diakses oleh admin melalui panel manajemen. Setelah laporan diterima, admin akan melakukan verifikasi terhadap kelengkapan dan validitas data yang dikirimkan. Jika laporan memenuhi kriteria yang telah ditetapkan, admin akan mengubah status laporan menjadi "Diproses". Sebaliknya, jika laporan tidak valid atau memerlukan revisi, admin dapat mengembalikannya atau "Ditolak" kepada pelapor untuk diperbaiki. Dapat dilihat pada Gambar 1.



Gambar 1. Proses admin

Flowchart proses pada pengguna

Pada Gambar 2 menjelaskan alur proses yang harus dilalui oleh pengguna dalam menggunakan sistem pelaporan kampus. Pada tahap awal, pengguna langsung disajikan formulir laporan untuk mengisi detail kejadian yang ingin dilaporkan. Pengguna harus mengisi laporan terlebih dahulu sebelum dapat mengakses sistem. Setelah formulir laporan diisi dengan lengkap, pengguna akan melanjutkan ke tahap pengiriman laporan ke *database*. Jika laporan berhasil dikirim, sistem akan mengalihkan pengguna ke halaman *login*, di mana pengguna harus memasukkan NIM dan Nama untuk melakukan autentikasi. Selanjutnya, sistem akan melakukan validasi data pengguna. Jika data yang dimasukkan *valid*, pengguna akan diarahkan ke dashboard, di mana pengguna dapat melihat status laporan yang telah dikirim. Namun, jika data yang dimasukkan tidak *valid*, pengguna akan menerima notifikasi gagal *login* dan diminta untuk mencoba kembali dengan informasi yang benar. Alur ini memastikan bahwa setiap pengguna hanya dapat mengakses sistem setelah terlebih dahulu mengirimkan laporan.



Gambar 2. Proses pengguna

Implementasi dan pengujian sistem

Setelah tahap perancangan sistem selesai dilakukan, sistem pelaporan kampus kemudian diimplementasikan dan diuji untuk memastikan kesesuaian antara perancangan dan hasil nyata di lapangan. Proses implementasi dilakukan menggunakan HTML, Tailwind CSS, JavaScript, dan PHP, dengan dukungan *database* MySQL untuk menyimpan data laporan mahasiswa. Proses implementasi berfokus pada realisasi antarmuka pengguna, halaman formulir pelaporan, halaman login pengguna, dashboard pengguna, halaman *login* admin, serta *dashboard* dan panel admin untuk pengelolaan laporan. Sementara itu, pengujian dilakukan dengan pendekatan *Black Box Testing* untuk menguji fungsionalitas sistem secara keseluruhan.

Halaman formulir pelaporan

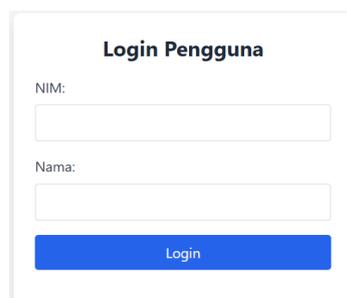
Pada Gambar 3 menampilkan antarmuka formulir pelaporan kampus yang dirancang untuk memfasilitasi mahasiswa dalam menyampaikan laporan terkait berbagai aspek di lingkungan kampus. Formulir ini terdiri dari beberapa bidang *input*, termasuk Nama, NIM, Kelas, Fakultas, Program Studi, Kategori, dan Deskripsi, yang bertujuan untuk mengidentifikasi pelapor secara jelas.



Gambar 3. Halaman formulir pelaporan

Halaman *login* pengguna

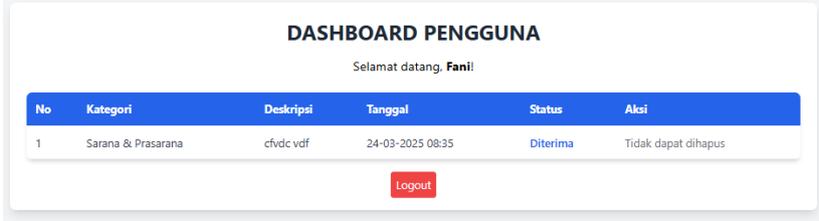
Gambar 4 menunjukkan antarmuka halaman *Login* Pengguna pada sistem pelaporan kampus. Halaman ini dirancang untuk memungkinkan pengguna mengakses *status* laporan yang telah dikirimkan sebelumnya. Untuk masuk ke dalam sistem, pengguna diharuskan mengisi NIM dan Nama sesuai dengan *form* laporan sebagai bentuk autentikasi. Halaman *login* ini akan muncul setelah pengguna menyelesaikan proses pelaporan dengan menekan tombol "Kirim Laporan" pada formulir pelaporan kampus. Dengan demikian, sistem memastikan bahwa hanya pengguna yang telah mengajukan laporan yang dapat mengakses dan memantau perkembangan laporan tersebut.



Gambar 4. Halaman login pengguna

Dashboard pengguna

Halaman ini berfungsi sebagai pusat informasi bagi pengguna untuk memantau status laporan yang telah dikirimkan. *Dashboard* ini menyajikan data laporan dalam bentuk tabel yang terdiri dari beberapa kolom, yaitu No, Kategori, Deskripsi, Tanggal, Status, dan Aksi. Untuk keluar dari sistem, terdapat tombol "*Logout*" berwarna merah yang memungkinkan pengguna mengakhiri sesi dengan aman.



The screenshot shows a dashboard titled "DASHBOARD PENGGUNA" with a welcome message "Selamat datang, Fani!". Below the message is a table with the following data:

No	Kategori	Deskripsi	Tanggal	Status	Aksi
1	Sarana & Prasarana	cfvdc vdf	24-03-2025 08:35	Diterima	Tidak dapat dihapus

Below the table is a red "Logout" button.

Gambar 5. *Dashboard* pengguna

Halaman login admin

Tampilan antarmuka *login* ini terdiri dari dua bidang *input*, yaitu *Username* dan *Password*, yang harus diisi oleh administrator untuk mendapatkan akses ke sistem. Setelah memasukkan kredensial yang benar, administrator dapat menekan tombol "*Login*" untuk masuk ke dalam sistem dan mengelola laporan pengguna.

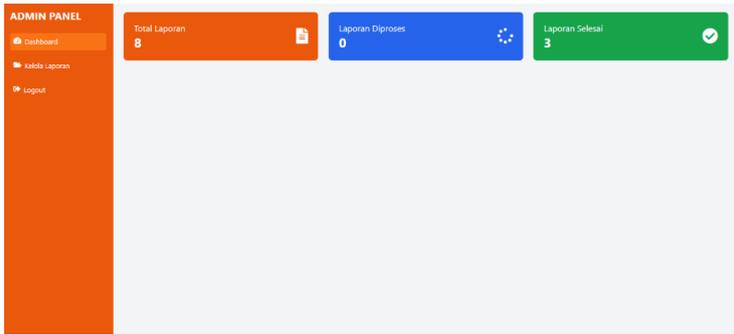


The screenshot shows a login form titled "Login Admin" with two input fields: "Username" and "Password". Below the fields is an orange "Login" button.

Gambar 6. Halaman login admin

Halaman dashboard admin

Dashboard ini berfungsi sebagai pusat kendali bagi administrator untuk memantau status laporan yang masuk dari pengguna. Pada bagian kiri layar, terdapat panel navigasi dengan tiga menu utama, yaitu *Dashboard*, *Kelola Laporan*, dan *Logout*, yang memudahkan administrator dalam mengakses berbagai fitur sistem.



The screenshot shows an admin dashboard with a navigation menu on the left and three summary cards at the top:

- ADMIN PANEL**
 - Dashboard
 - Kelola Laporan
 - Logout
- Total Laporan**: 8
- Laporan Diproses**: 0
- Laporan Selesai**: 3

Gambar 7. *Dashboard* admin

Halaman daftar laporan

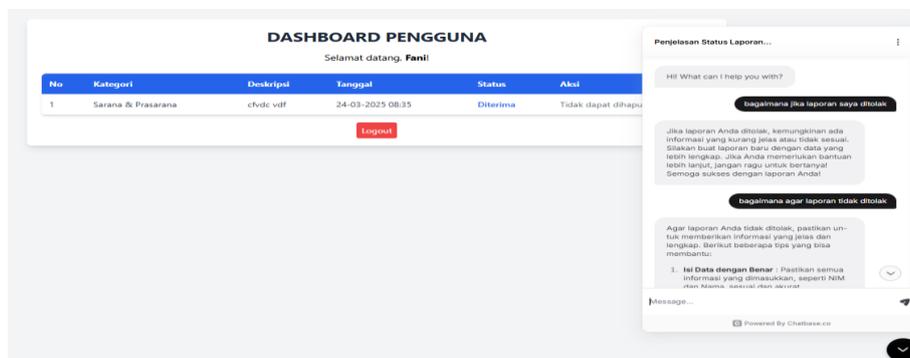
Halaman ini berfungsi sebagai pusat pengelolaan berbagai laporan yang telah diajukan oleh pengguna. Setiap entri dalam tabel memuat informasi penting seperti nomor, nama pelapor, NIM, kelas, fakultas, program studi, kategori laporan, deskripsi laporan, tanggal pengajuan, status laporan, dan opsi aksi. Kolom "Status" menunjukkan laporan telah diproses atau tidak, sedangkan kolom "Aksi" menyediakan tombol "Ubah" dan "Hapus" untuk memungkinkan admin memodifikasi atau menghapus laporan.

No	Nama Pelapor	NIM	Kelas	Fakultas	Program Studi	Kategori	Deskripsi	Tanggal	Status	Aksi
1	test	0	test	test	test	Sarana & Prasarana	test	24-03-2025 23:51	Diterima	Diterima Ubah Hapus
2	Bebek	2345	v	fvxc vfd	sdcfrcdf	Sarana & Prasarana	dfvfgvfd	24-03-2025 18:28	Diterima	Diterima Ubah Hapus
3	Freyro	4233590033	c	sw	xss	Sarana & Prasarana	csxc	24-03-2025 08:39	Diterima	Diterima Ubah Hapus
4	Fani	310106	caddc	c ds	cf fv	Sarana & Prasarana	cfvdc vdf	24-03-2025 08:35	Diterima	Diterima Ubah Hapus
5	Berli	123	cdffwcds	xcsd s	csd vf	Sarana & Prasarana	c svfc	24-03-2025 08:26	Diterima	Diterima Ubah Hapus
6	franco	5323467587	fvxc fvgtdfbav	dv gbthbvdfs	sfdvgedfvzx	Sarana & Prasarana	dc bdg vvd	22-03-2025 23:06	Selesai	Selesai Ubah Hapus

Gambar 8. Halaman daftar laporan

Penerapan chatbot

Fitur *chatbot* ini diterapkan pada *dashboard* pengguna dan halaman *login* pengguna. *Chatbot* ini secara otomatis merespons pertanyaan pengguna mengenai proses pelaporan. Jika membutuhkan bantuan lebih lanjut, pengguna dapat mengajukan pertanyaan tambahan. Kehadiran *chatbot* ini mempermudah pengguna memahami proses pelaporan dan membantu memberikan informasi yang dibutuhkan pengguna.



Gambar 9. Tampilan penerapan chatbot

Hasil pengujian sistem

Pengujian dilakukan untuk memastikan bahwa fitur-fitur yang telah diimplementasikan berjalan dengan baik dan sesuai spesifikasi yang ditentukan. Pengujian dilakukan dengan metode *Black Box Testing*, di mana setiap fungsi diuji berdasarkan masukan dan keluaran tanpa melihat struktur internal kode program. Dapat dilihat pada tabel 1.

Tabel 1. Hasil pengujian

<i>No</i>	<i>Fitur yang diuji</i>	<i>Skenario Uji</i>	<i>Hasil yang diharapkan</i>	<i>Hasil Pengujian</i>
1	Formulir Pelaporan	Mahasiswa mengisi dan mengirim laporan	Data tersimpan ke <i>database</i>	Berhasil
2	Status Laporan	Mahasiswa cek <i>status</i> dengan Nim dan Nama	Status tampil sesuai laporan	Berhasil
3	Panel Admin	Admin melihat dan ubah <i>status</i> laporan	Status berhasil diperbarui	Berhasil
4	<i>Chatbot</i>	<i>Chatbot</i> ditanya prosedur pelaporan	Balasan otomatis muncul	Berhasil

4. KESIMPULAN DAN SARAN

Penelitian ini telah berhasil merancang dan mengembangkan aplikasi *web* pelaporan kampus di Universitas Negeri Medan yang terintegrasi dengan fitur *chatbot*. Aplikasi ini dibangun dengan menggunakan teknologi *web* seperti HTML, *Tailwind* CSS, JavaScript, dan PHP, serta didukung oleh MySQL sebagai *basis data*. Berdasarkan proses implementasi dan pengujian yang telah dilakukan, sistem terbukti mampu mendukung proses pelaporan secara digital dengan menyediakan fitur formulir pelaporan, autentikasi pengguna, *dashboard status* laporan, serta *panel admin* untuk pengelolaan laporan. Selain itu, *chatbot* yang terintegrasi berperan penting dalam memberikan informasi secara otomatis kepada pengguna, sehingga meningkatkan efisiensi dan kenyamanan dalam proses pelaporan. Namun demikian, masih terdapat beberapa keterbatasan yang perlu diperhatikan. Salah satunya adalah keterbatasan *chatbot* dalam merespons pertanyaan yang lebih kompleks dan tidak terstruktur. Selain itu, pengujian sistem masih dilakukan dalam skala terbatas dan belum mencakup simulasi beban tinggi. Oleh karena itu, untuk pengembangan selanjutnya, disarankan agar dilakukan pengujian dengan partisipasi pengguna yang lebih luas dan kondisi operasional yang lebih realistis.

5. DAFTAR PUSTAKA

- [1] R. Dwi Cahyani, Mohammad Ravtayya Attaturk Maklin, Syaila Ananta Karenina, Salma Ashiila Rabbani, Aruni Intan Haifa Sumawidjaya, and R. Setyadinsa, "Perancangan Aplikasi Mobile 'ProtectU' untuk Pelaporan Kekerasan Seksual di Lingkungan Universitas Berbasis Android Studio," *Inform. J. Ilmu Komput.*, vol. 20, no. 1, pp. 16–27, 2024, doi: 10.52958/iftk.v20i1.8150.
- [2] A. F. Setianda and N. D. Saputro, "Aplikasi Sistem Informasi Pengaduan Mahasiswa Di UPT TIK Universitas PGRI Semarang," *Sci. Eng. Natl. Semin.* 5

- (SENS 5), vol. 5, no. Sens 5, pp. 679–685, 2020.
- [3] F. Sahfitri, A. Iefan, D. I. B. Neo, K. Amadea, and P. Andhika, “Perancangan UI / UX Website Promosi Energi Terbarukan Panel Surya Dengan Metode Agile Development (STUDI KASUS CV . PAMA SOLAR),” pp. 241–253, 2024. *Jurnal Teknologi Informasi dan Komputer*, 10(4).
- [4] F. Sinlae, I. Maulana, F. Setiyansyah, and M. Ihsan, “Pengenalan Pemrograman Web: Pembuatan Aplikasi Web Sederhana Dengan PHP dan MYSQL,” *J. Siber Multi Disiplin*, vol. 2, no. 2, pp. 68–82, 2024, [Online]. Available: <https://creativecommons.org/licenses/by/4.0/>
- [5] D. Ramadaniah and J. Fitra, “Rekayasa Perangkat Lunak: Aplikasi Perpustakaan Menggunakan Framework CodeIgniter,” *J. Softw. Eng. Technol.*, pp. 51–58, 2021.
- [6] I. Noval Niati, L. Hadjaratie, and T. Abdillah, “Sistem Informasi Pelayanan Pengaduan Mahasiswa Berbasis Android,” *Diffusion*, vol. 3, no. 1, pp. 14–26, 2023, [Online]. Available: <https://ejurnal.ung.ac.id/index.php/diffusion/article/view/12406/5962>
- [7] I. S. Nugroho and A. Voutama, “Implementasi Chat Bot Untuk Pelayanan Pelanggan Yang Terintegrasi Web Toko Komputer,” *JATI (Jurnal Mhs. Tek. Inform.)*, vol. 8, no. 3, pp. 3132–3136, 2024, doi: 10.36040/jati.v8i3.9630.
- [8] B. Huberta and A. B. Wijaya, “Perancangan Chatbot Website Program Studi Informatika Menggunakan Framework Codeigniter,” *J. Inform. dan Tek. Elektro Terap.*, vol. 11, no. 3, 2023, doi: 10.23960/jitet.v11i3.3225.
- [9] A. N. A. Saimona, B. Kurniawan, and D. S. Agustina, “Sistem Informasi Penjualan Pada Toko Anugrah Setia Berbasis Web Menggunakan PHP dan MYSQL,” *JTIM J. Tek. Inform. Mahakarya*, vol. 03, no. 2, pp. 37–44, 2020.
- [10] A. H. Fathulloh and H. I. Adauwiyah, “Perbandingan Tingkat Efisiensi Waktu Query SELECT pada Database Interface Navicat dan SQLYog di MySQL DBMS,” *Appl. Inf. Syst. Manag.*, vol. 4, no. 2, pp. 101–105, 2021, doi: 10.15408/aism.v4i2.18369.