



Pengembangan Sistem Informasi Penanganan Feedback Client Berbasis Web Menggunakan Metode Waterfall

Caesar Yoga Pratama*, Marta Ardiyanto, Mira Erlinawati

Fakultas Ilmu Komputer, Program Studi Sistem Informasi, Universitas Duta Bangsa Surakarta, Surakarta, Indonesia

Email: ^{1,*}220101048@mhs.udb.ac.id, ²marta.ardiyanto@udb.ac.id, ³mira_erlinawati@udb.ac.id

Email Penulis Korespondensi: 220101048@mhs.udb.ac.id

Abstrak—Pengelolaan *feedback client* di BIIS Corp masih mengandalkan grup *WhatsApp* dan pencatatan manual melalui *spreadsheet*, sehingga menimbulkan kendala seperti sulitnya memantau progres pengerjaan, minimnya dokumentasi riwayat penanganan, serta lemahnya koordinasi antara *staff* dan *programmer*. Penelitian ini bertujuan merancang dan membangun sistem informasi *feedback client* berbasis web yang mampu mendukung pengelolaan *feedback* secara terpusat dan terstruktur. Identifikasi masalah dilakukan menggunakan *framework* PIECES yang mengungkap kelemahan pada keenam aspeknya. Pengembangan sistem mengacu pada model *Waterfall* dalam kerangka SDLC, dengan pemodelan menggunakan UML serta pengujian fungsional melalui metode *Blackbox Testing*. Sistem dibangun menggunakan PHP, MySQL, HTML, CSS, dan JavaScript, serta mengakomodasi tiga peran pengguna yaitu *admin*, *staff*, dan *programmer*, dilengkapi fitur pencatatan *feedback*, penugasan *programmer*, pembaruan status, konfirmasi penyelesaian, mekanisme *reopen feedback*, notifikasi *real-time*, *activity log*, dan laporan kinerja. Pengujian terhadap 15 skenario menghasilkan tingkat keberhasilan 100%. Kontribusi penelitian ini adalah menghasilkan sistem yang dirancang khusus untuk lingkungan *software house* dengan mengintegrasikan ketiga peran pengguna dalam satu alur kerja terpusat yang dilengkapi mekanisme *reopen feedback*, notifikasi *real-time*, dan *activity log* untuk mendukung proses monitoring dan dokumentasi penanganan *feedback* secara lebih efektif.

Kata Kunci: Sistem Informasi; Feedback Client; PIECES Framework; Waterfall; Blackbox Testing

Abstract—Client feedback management at BIIS Corp still relies on WhatsApp group communication and manual spreadsheet recording, resulting in difficulties in monitoring work progress, inadequate documentation of handling history, and poor coordination between staff and programmers. This study aims to design and develop a web-based client feedback information system to support centralized and structured feedback management. Problem identification was conducted using the PIECES framework, which revealed weaknesses across all six aspects. The system was developed using the Waterfall model within the SDLC framework, with UML-based modeling and functional testing through the Blackbox Testing method. Built using PHP, MySQL, HTML, CSS, and JavaScript, the system accommodates three user roles, namely administrator, staff, and programmer, and is equipped with features including feedback recording, programmer assignment, status updates, completion confirmation, reopen feedback mechanism, real-time notifications, activity log, and performance reporting. Testing across 15 scenarios achieved a 100% success rate. The contribution of this research is the development of a system specifically designed for a software house environment, integrating three user roles into a centralized workflow equipped with a reopen feedback mechanism, real-time notifications, and an activity log to support more effective feedback monitoring and documentation.

Keywords: Information System; Client Feedback; PIECES Framework; Waterfall; Blackbox Testing

1. PENDAHULUAN

Dalam perusahaan pengembang perangkat lunak, pengelolaan *feedback client* merupakan bagian penting dalam menjaga kualitas layanan kepada client. Proses ini mencakup penyampaian laporan, pemantauan progres perbaikan, serta koordinasi antar tim pengembang untuk memastikan setiap kebutuhan client dapat ditangani secara tepat. Pengelolaan *feedback* yang masih dilakukan secara manual atau melalui media komunikasi yang tidak terstruktur berpotensi menyebabkan keterlambatan respons, kesulitan pelacakan status, dan tidak terdokumentasinya riwayat penanganan secara baik. Penelitian yang dilakukan (Mufarikuddin & Surianto, 2022) menjelaskan bahwa penanganan keluhan pelanggan yang dilakukan secara tepat dan didukung oleh layanan yang baik dapat meningkatkan efektivitas pelayanan serta membangun kepercayaan pelanggan terhadap perusahaan.

BIIS Corp adalah sebuah perusahaan yang fokus pada layanan pengembangan *software* berbasis web dan melayani berbagai kebutuhan sistem informasi bagi para *client*nya. Dalam proses pengembangan dan pemeliharaan sistem, *feedback* dari *client* menjadi bagian penting untuk memastikan perbaikan maupun penyesuaian sistem dapat dilakukan sesuai kebutuhan pengguna. Saat ini proses penyampaian *feedback* di BIIS Corp masih dilakukan melalui media komunikasi seperti *WhatsApp*, kemudian dicatat kembali oleh *staff* ke dalam *spreadsheet* sebelum diteruskan kepada *programmer* untuk ditindaklanjuti. *Feedback* yang diterima umumnya berkaitan dengan perbaikan *bug*, perubahan tampilan sistem, penyesuaian fitur, maupun permintaan pengembangan tambahan sesuai kebutuhan pengguna. Oleh karena itu, proses pengelolaan *feedback* memerlukan pencatatan dan dokumentasi yang baik agar setiap tindak lanjut yang dilakukan *programmer* dapat dipantau secara lebih jelas dan terorganisir. Proses tersebut membantu koordinasi antara *client* dan tim pengembang, namun penggunaan media komunikasi dan pencatatan yang masih terpisah menyebabkan proses monitoring *feedback* menjadi kurang terstruktur. Penelitian (Muhyiddin dkk., 2023) menjelaskan bahwa pengelolaan umpan balik pelanggan yang masih mengandalkan media sosial dan *spreadsheet* menyebabkan proses penanganan menjadi kurang efektif, data tidak terdokumentasi secara optimal, serta memperlambat koordinasi antarbagian. Oleh karena itu, pengembangan sistem pengelolaan *feedback* diperlukan untuk mendukung proses pencatatan, monitoring, dan tindak lanjut yang lebih terstruktur. Penelitian (Zunindar & Wiyono,



2023) menjelaskan bahwa proses pengaduan yang masih dilakukan secara manual berisiko menimbulkan keterlambatan respons, kesulitan pencarian data, serta kendala dalam pengelolaan laporan secara efektif.

Pengelolaan *feedback client* dalam proses pengembangan perangkat lunak memerlukan sistem berbasis web agar seluruh proses pencatatan, penugasan, pemantauan status, dan dokumentasi riwayat penanganan dapat dilakukan secara terpusat serta diakses secara *real-time*. Dengan mekanisme tersebut, staff dan programmer dapat memantau perkembangan setiap *feedback*, mengurangi risiko kehilangan informasi, serta meningkatkan efektivitas koordinasi selama proses penyelesaian. (Damayanti dkk., 2023) menjelaskan bahwa sistem pengaduan berbasis web mampu membantu proses pelayanan dan mempermudah pengguna dalam menyampaikan maupun mengelola laporan secara lebih efektif. Selain itu, (Hendry dkk., 2025) menyatakan bahwa penerapan mekanisme *automatic ticketing workflow* dan pelacakan status tiket dapat membantu meningkatkan transparansi proses, mempercepat tindak lanjut laporan, serta mendukung pengelolaan laporan secara lebih terstruktur.

Penelitian mengenai sistem *helpdesk* dan *ticketing* berbasis web telah banyak dilakukan untuk membantu proses pengelolaan laporan dan monitoring penanganan masalah secara lebih terstruktur. (Sofyan dkk., 2024) mengembangkan sistem *ticketing* berbasis website pada PT. Qonita Teknologi Indonesia yang mampu mendukung pembuatan tiket layanan, monitoring status tiket, serta mendokumentasikan penyelesaian masalah secara terpusat. Penelitian lain yang dilakukan oleh (Rosmalina dkk., 2025) mengembangkan sistem informasi keluhan pelanggan berbasis Laravel untuk membantu proses pengajuan keluhan dan pelacakan status penyelesaian secara *real-time* sehingga dapat meningkatkan transparansi layanan pelanggan. Selain itu, (Tarigan dkk., 2022) menjelaskan bahwa penerapan *helpdesk ticketing system* dapat membantu proses monitoring penanganan keluhan, pencatatan histori masalah, serta pengelolaan laporan secara lebih efektif dan terstruktur. Penelitian lain oleh (Irawan dkk., 2024) menunjukkan bahwa penerapan sistem informasi proyek berbasis web dapat mendukung monitoring status pekerjaan secara *real-time*, sehingga koordinasi antar pengguna menjadi lebih terstruktur dan proses pelacakan pekerjaan dapat dilakukan dengan lebih efektif. Penelitian yang dilakukan oleh (Pratomo, 2023) menunjukkan bahwa penerapan website manajemen tugas berbasis web yang dilengkapi fitur penugasan, monitoring progres, dan notifikasi otomatis mampu meningkatkan koordinasi antar pengguna serta mendukung pengelolaan pekerjaan secara lebih baik.

Berdasarkan beberapa penelitian terdahulu, sistem *helpdesk* dan *ticketing* berbasis web telah banyak dikembangkan untuk membantu proses pengelolaan laporan, monitoring penanganan keluhan, serta pelacakan status pekerjaan secara lebih terstruktur. Namun, penelitian yang berfokus pada pengelolaan *feedback client* dalam proses pengembangan perangkat lunak di lingkungan *software house* masih memerlukan penyesuaian terhadap alur koordinasi yang berjalan di perusahaan. Pada BIIS Corp, proses *feedback* melibatkan *staff* sebagai penghubung antara *client* dan *programmer* sehingga dibutuhkan sistem yang mampu membantu proses pencatatan *feedback*, monitoring status pekerjaan, serta dokumentasi riwayat penanganan *feedback* secara terpusat. Berdasarkan kondisi tersebut, penelitian ini diarahkan untuk membangun sebuah sistem informasi berbasis web di lingkungan BIIS Corp yang dapat mengoptimalkan proses pengelolaan *feedback client* sehingga aktivitas monitoring dan tindak lanjut dapat berlangsung lebih terstruktur. Kontribusi utama penelitian ini terletak pada pengembangan sistem informasi penanganan *feedback client* yang disesuaikan dengan alur kerja *software house*, yaitu mengintegrasikan proses pencatatan *feedback*, penugasan programmer, monitoring status, mekanisme reopen *feedback*, notifikasi *real-time*, dan pencatatan *activity log* dalam satu platform terpusat. Kontribusi tersebut diharapkan dapat mendukung koordinasi antar pengguna serta meningkatkan efektivitas pengelolaan *feedback* di BIIS Corp.

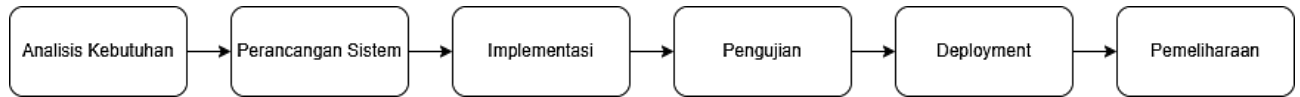
2. METODOLOGI PENELITIAN

2.1 Kerangka Dasar Penelitian

Penelitian ini dilaksanakan melalui alur kerja yang terstruktur dan bertahap, mencakup identifikasi masalah, pengumpulan data, analisis kebutuhan, perancangan, implementasi, hingga pengujian sistem. Data diperoleh melalui observasi lapangan, wawancara dengan pihak yang terlibat, serta kajian pustaka yang relevan, guna mendapatkan pemahaman mendalam mengenai kondisi pengelolaan *feedback client* yang sedang berjalan di perusahaan. Penyusunan tahapan penelitian yang sistematis berperan penting dalam memastikan proses pengembangan berjalan terarah dan selaras dengan kebutuhan pengguna (Fachri dkk., 2024; Fuaddin dkk., 2025). Untuk mengidentifikasi kelemahan pada sistem yang sedang berjalan, analisis dilakukan menggunakan *PIECES Framework* yang mengevaluasi sistem dari enam dimensi, yaitu *performance*, *information*, *economy*, *control*, *efficiency*, dan *service* (Fuaddin dkk., 2025; Maulana & Budiyanantara, 2025). Hasil analisis kemudian dijadikan dasar dalam perancangan sistem, yang dituangkan ke dalam pemodelan menggunakan *Unified Modelling Language (UML)* untuk menggambarkan alur dan kebutuhan sistem secara terstruktur. Pembangunan sistem mengadopsi metode *Waterfall* yang dikenal memiliki alur pekerjaan sekuensial dan terorganisir, mulai dari tahap analisis hingga pengujian akhir (Anis dkk., 2024; Fachri dkk., 2024). Sebagai tahap akhir, pengujian sistem dilakukan dengan menerapkan metode *Blackbox Testing* guna memverifikasi bahwa setiap fungsi sistem menghasilkan keluaran yang sesuai dengan kebutuhan pengguna dan rancangan yang telah ditetapkan (Silitonga dkk., 2023; Zen dkk., 2024). Penjelasan mengenai setiap tahapan penelitian, mulai dari analisis kebutuhan menggunakan *framework* *PIECES*, perancangan sistem menggunakan *UML*, implementasi berdasarkan model *Waterfall*, hingga pengujian menggunakan *Blackbox Testing* dijelaskan pada subbab berikutnya.

2.2 Tahapan Penelitian

Dalam penelitian ini, pendekatan pengembangan yang dipilih adalah metode *Waterfall*, mengingat metode ini menawarkan alur kerja yang terurut dan terstruktur mulai dari tahap penggalan kebutuhan sampai dengan pengujian akhir sistem. Rangkaian tahapan yang ditempuh dalam penelitian ini dapat diamati pada Gambar 1.



Gambar 1. Tahapan Penelitian

Berdasarkan Gambar 1, metode *Waterfall* terdiri atas enam tahapan yang dilaksanakan secara berurutan, yaitu analisis kebutuhan, perancangan sistem, implementasi, pengujian, *deployment*, dan pemeliharaan. Setiap tahapan memiliki keterkaitan sehingga hasil dari suatu tahap menjadi dasar pelaksanaan tahap berikutnya. Pada penelitian ini, seluruh tahapan tersebut diterapkan sebagai pedoman dalam proses pengembangan sistem informasi penanganan *feedback client* di BIIS Corp.

2.2.1 Analisis Kebutuhan

Tahap analisis kebutuhan difokuskan pada pemetaan permasalahan serta kebutuhan sistem berdasarkan kondisi pengelolaan *feedback client* yang sedang berjalan di BIIS Corp. Untuk mendapatkan gambaran yang menyeluruh, pengumpulan data dilakukan melalui tiga cara, yaitu observasi langsung, wawancara dengan pihak yang terlibat, dan penelusuran literatur terkait.

2.2.2 Perancangan Sistem

Berdasarkan temuan dari tahap analisis, selanjutnya dilakukan perancangan sistem yang mencakup penggambaran alur dan kebutuhan fungsional. Tahap ini menghasilkan sejumlah artefak perancangan berupa diagram UML, skema *database*, serta rancangan antarmuka yang menjadi panduan teknis dalam tahap implementasi.

2.2.3 Implementasi

Pada tahap implementasi, seluruh hasil rancangan diwujudkan ke dalam bentuk aplikasi web yang siap digunakan. Kegiatan pada tahap ini mencakup pembangunan fitur-fitur sistem, pengelolaan struktur *database*, serta pengembangan antarmuka pengguna yang disesuaikan dengan kebutuhan operasional BIIS Corp.

2.2.4 Pengujian

Setelah proses pembangunan sistem rampung, dilakukan pengujian menggunakan metode *Blackbox Testing*. Pengujian direncanakan terhadap 15 *skenario* yang mencakup fungsi-fungsi utama sistem, meliputi proses *login*, pengelolaan *feedback*, penugasan programmer, pembaruan status, konfirmasi penyelesaian, mekanisme *reopen feedback*, pengelolaan data pengguna, notifikasi, *activity log*, serta pembuatan laporan. Pengujian dilakukan untuk memverifikasi bahwa setiap fungsi menghasilkan keluaran yang sesuai dengan kebutuhan fungsional dan alur sistem yang telah dirancang.

2.2.5 Deployment

Sistem yang telah melewati tahap pengujian kemudian didistribusikan kepada BIIS Corp dengan menyerahkan aplikasi beserta basis data untuk diimplementasikan pada lingkungan operasional perusahaan. Selanjutnya, pengguna dapat mengakses sistem sesuai dengan peran dan hak akses yang telah ditetapkan.

2.2.6 Pemeliharaan

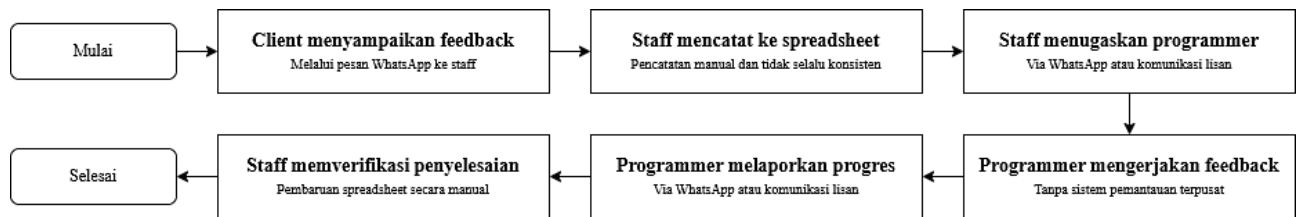
Guna memastikan sistem tetap berfungsi optimal pasca penerapan, dilakukan tahap pemeliharaan secara berkala. Kegiatan ini meliputi penanganan *bug*, penyesuaian terhadap perubahan kebutuhan pengguna, serta penambahan fitur baru apabila dibutuhkan dalam penggunaan sistem ke depannya.

3. HASIL DAN PEMBAHASAN

3.1 Analisis Sistem yang sedang Berjalan

Sebelum merancang sistem yang baru, dilakukan analisis terhadap proses pengelolaan *feedback client* yang saat ini berjalan di BIIS Corp. Analisis ini bertujuan untuk memahami alur proses penanganan *feedback*, mengetahui mekanisme koordinasi antara *staff* dan *programmer* serta mengidentifikasi kendala yang terjadi dalam proses pengelolaan *feedback client*. Informasi yang digunakan dalam analisis sistem berjalan diperoleh melalui observasi dan wawancara secara langsung dengan *staff* dan *programmer* BIIS Corp yang terlibat dalam proses penanganan *feedback client*. Berdasarkan hasil observasi, proses pengelolaan *feedback* di BIIS Corp saat ini masih dilakukan secara manual dan belum menggunakan sistem informasi yang terintegrasi. Penyampaian *feedback* dari *client* dilakukan melalui

aplikasi *WhatsApp*, kemudian *staff* melakukan pencatatan *feedback* ke dalam *spreadsheet* sebelum diteruskan kepada programmer untuk ditindaklanjuti. Seluruh proses koordinasi pengerjaan *feedback*, pelaporan progres, hingga konfirmasi penyelesaian masih dilakukan melalui komunikasi manual menggunakan *WhatsApp*. Alur sistem yang berjalan dapat dilihat pada Gambar 2.



Gambar 2. Analisis sistem yang sedang berjalan

Berdasarkan analisis pada Gambar 2, proses dimulai ketika *client* menyampaikan *feedback* kepada *staff* melalui pesan *WhatsApp*. *Feedback* yang diterima kemudian dicatat secara manual oleh *staff* ke dalam *spreadsheet* sebagai dokumentasi sementara. Setelah itu, *staff* meneruskan *feedback* tersebut kepada *programmer* melalui komunikasi *WhatsApp* atau komunikasi secara langsung untuk dilakukan proses perbaikan maupun penyesuaian sistem sesuai kebutuhan *client*. Selanjutnya *Programmer* mulai mengerjakan *feedback* yang telah diterima tanpa adanya sistem pemantauan terpusat yang dapat digunakan untuk memonitor progres pengerjaan secara langsung. Setelah proses pengerjaan dilakukan, *programmer* melaporkan progres maupun hasil pengerjaan kepada *staff* melalui *WhatsApp*. *Staff* kemudian melakukan verifikasi terhadap *feedback* yang telah diselesaikan dan memperbarui data pada *spreadsheet* secara manual sebelum hasil tindak lanjut disampaikan kembali kepada *client*.

Proses pengelolaan *feedback* yang masih dilakukan secara manual menyebabkan proses monitoring pengerjaan *feedback* belum berjalan secara optimal. Selain itu, penggunaan media komunikasi dan pencatatan yang terpisah menyebabkan riwayat penanganan *feedback* belum terdokumentasi secara terpusat sehingga proses pelacakan status pengerjaan dan pencarian data *feedback* tertentu masih membutuhkan koordinasi tambahan antara *staff* dan *programmer*.

3.2 Analisis Kelemahan Sistem

Berdasarkan hasil observasi dan wawancara, kondisi sistem yang sedang berjalan kemudian dianalisis menggunakan kerangka PIECES untuk mengidentifikasi permasalahan secara lebih terstruktur. Hasil analisis disajikan pada Tabel 1 berikut:

Tabel 1. Analisis PIECES

Aspek	Permasalahan	Dampak
<i>Performance</i>	Proses penerimaan dan penerusan <i>feedback</i> bergantung pada notifikasi pesan <i>WhatsApp</i> yang bercampur dengan percakapan lain, sehingga tidak ada mekanisme prioritas atau batas waktu penanganan yang jelas.	Waktu respons <i>feedback client</i> menjadi tidak teratur dan berpotensi tidak konsisten antar proyek, tergantung pada seberapa cepat <i>staff</i> menyadari adanya pesan baru.
<i>Information</i>	Data <i>feedback</i> tersebar di dua media yang berbeda, yaitu percakapan <i>WhatsApp</i> dan <i>spreadsheet</i> yang tidak selalu diperbarui secara konsisten. Tidak ada format pencatatan yang seragam untuk setiap <i>feedback</i> yang masuk.	Informasi yang tersedia sering kali tidak lengkap, sehingga sulit untuk mengetahui status penanganan <i>feedback</i> secara akurat tanpa harus menghubungi pihak terkait secara langsung.
<i>Economy</i>	<i>Staff</i> menghabiskan waktu untuk mencari kembali <i>feedback</i> di riwayat percakapan, memperbarui <i>spreadsheet</i> secara manual, dan melakukan tindak lanjut melalui komunikasi yang terpisah-pisah.	Waktu kerja <i>staff</i> dan <i>programmer</i> tidak dimanfaatkan secara optimal karena sebagian digunakan untuk koordinasi administratif yang bersifat repetitif dan dapat diotomatiskan.
<i>Control</i>	Tidak ada sistem penugasan formal yang mencatat siapa bertanggung jawab atas <i>feedback</i> tertentu. Pembaruan status pengerjaan sepenuhnya bergantung pada inisiatif individu tanpa mekanisme peringatan atau eskalasi.	Akuntabilitas penanganan <i>feedback</i> menjadi tidak jelas. Ketika <i>feedback</i> tidak tertangani, sulit untuk melacak pada tahap mana proses terhenti dan siapa yang seharusnya bertindak.
<i>Efficiency</i>	Proses penanganan melibatkan perpindahan antar alat yang tidak terintegrasi, penerimaan <i>feedback</i> di <i>WhatsApp</i> , pencatatan di <i>spreadsheet</i> , penugasan via <i>WhatsApp</i> atau	<i>Programmer</i> tidak memiliki tampilan tugas yang terpusat, sehingga harus mengandalkan memori atau bertanya secara langsung kepada <i>staff</i> untuk mengetahui daftar <i>feedback</i> yang perlu dikerjakan.

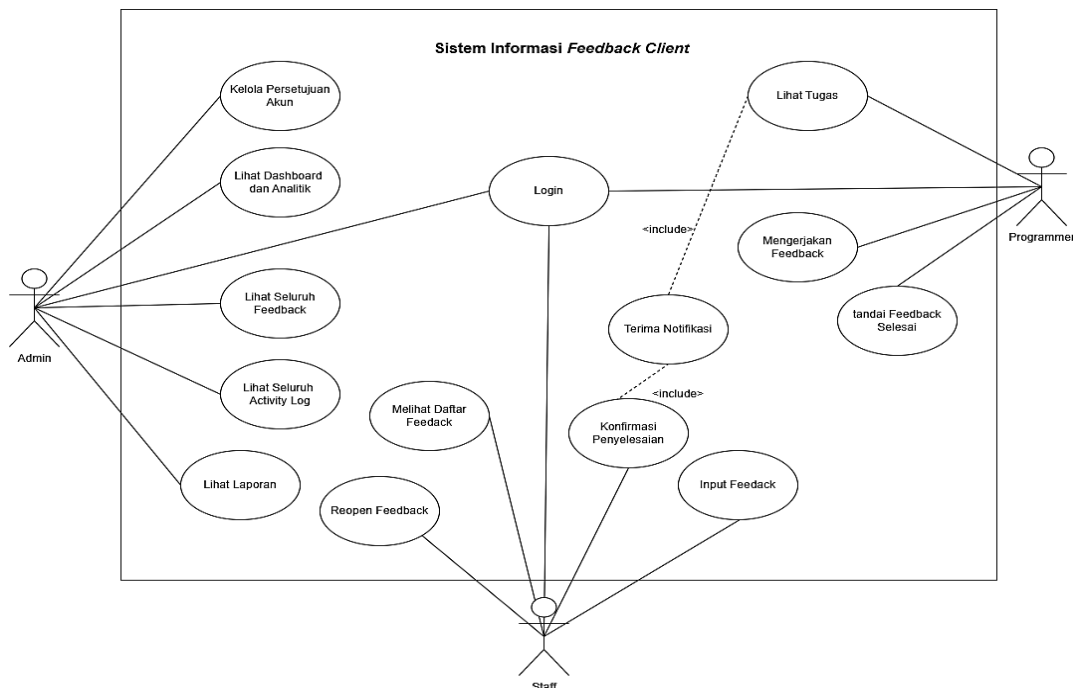
Aspek	Permasalahan	Dampak
	lisan, dan pelaporan progres kembali via <i>WhatsApp</i> .	
Service	Tidak tersedia mekanisme notifikasi otomatis ketika terjadi perubahan status <i>feedback</i> , baik untuk <i>programmer</i> yang menerima tugas baru maupun untuk <i>staff</i> yang menunggu konfirmasi penyelesaian.	<i>Staff</i> harus secara aktif menghubungi <i>programmer</i> untuk mendapatkan pembaruan status, yang berpotensi memperlambat proses konfirmasi dan menyulitkan koordinasi terutama saat banyak <i>feedback</i> berjalan bersamaan.

Berdasarkan hasil analisis PIECES di atas, dapat disimpulkan bahwa sistem yang berjalan saat ini memiliki kelemahan pada seluruh aspek yang dianalisis. Permasalahan yang paling menonjol adalah ketiadaan mekanisme pencatatan dan pemantauan yang terpusat, sehingga koordinasi antar pihak yang terlibat menjadi tidak efisien dan riwayat penanganan *feedback* tidak terdokumentasi dengan baik. Kondisi inilah yang menjadi landasan utama diperlukannya pengembangan sistem informasi yang terintegrasi.

3.3 Perancangan Sistem

3.3.1 Use Case Diagram

Terdapat tiga aktor yang berinteraksi dalam sistem yang dibangun, yaitu *Admin*, *Staff*, dan *Programmer*. Relasi antara masing-masing aktor dengan fungsionalitas sistem secara keseluruhan direpresentasikan melalui *use case diagram* yang ditampilkan pada Gambar 3.



Gambar 3. Use Case Diagram

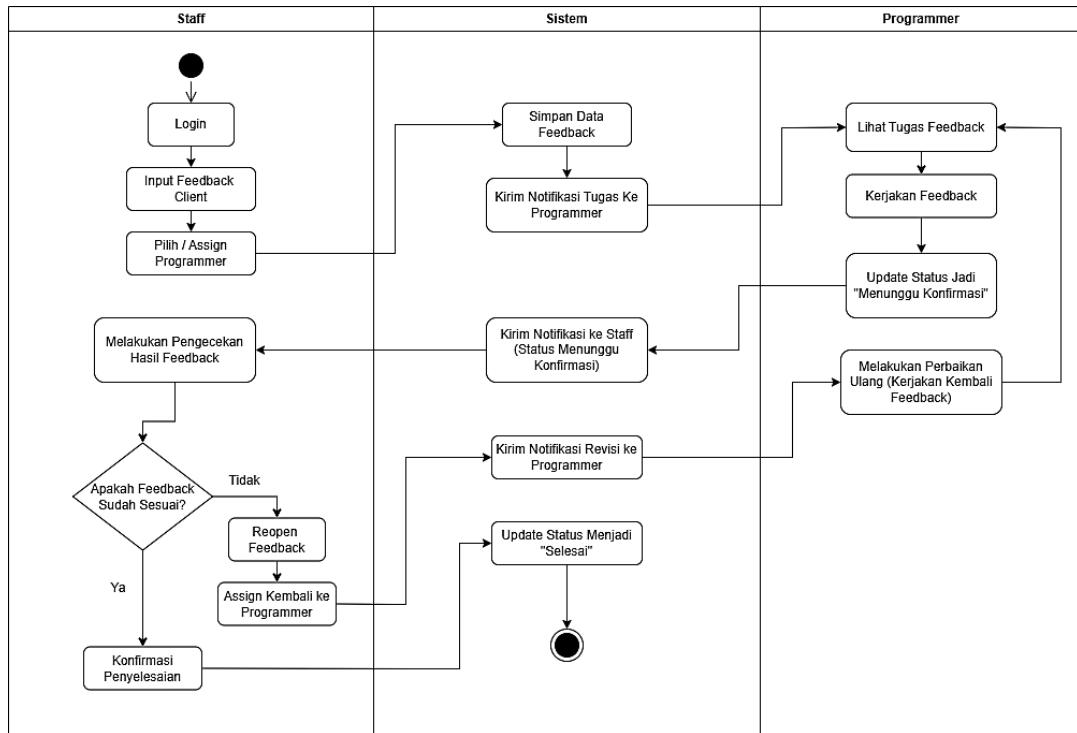
Berdasarkan Gambar 3, admin memiliki akses terhadap enam *use case*, meliputi login, kelola persetujuan akun, lihat *dashboard* dan analitik, lihat seluruh *feedback*, lihat *activity log*, serta lihat laporan. *Staff* berinteraksi dengan lima *use case*, yaitu login, *input feedback*, lihat daftar *feedback*, konfirmasi penyelesaian, dan *reopen feedback*. *Programmer* memiliki empat *use case*, yaitu login, lihat tugas, kerjakan *feedback*, dan tandai *feedback* selesai. Selain itu, *staff* dan *programmer* sama-sama memiliki akses terhadap *use case* terima notifikasi yang digunakan sebagai mekanisme pemberitahuan secara *real-time* setiap kali perubahan status *feedback*.

3.3.2 Activity Diagram

Untuk memvisualisasikan alur kerja pengelolaan *feedback client* dalam sistem yang dirancang, digunakan *activity diagram*. Diagram ini menguraikan rangkaian aktivitas yang melibatkan *staff*, sistem, dan *programmer* sejak proses *input feedback* hingga konfirmasi penyelesaian, sebagaimana ditampilkan pada Gambar 4.

Berdasarkan Gambar 4, proses dimulai ketika *staff* melakukan login ke dalam sistem dan menginput *feedback client*. Setelah data *feedback* diinput, *staff* melakukan *assign programmer* yang bertugas menangani *feedback* tersebut. Sistem kemudian menyimpan data *feedback* dan mengirimkan notifikasi tugas kepada *programmer*. Selanjutnya *programmer* melihat tugas *feedback* yang diterima dan mulai melakukan proses pengerjaan *feedback*. Setelah proses pengerjaan selesai dilakukan, *programmer* memperbarui status *feedback* menjadi “Menunggu Konfirmasi” sehingga

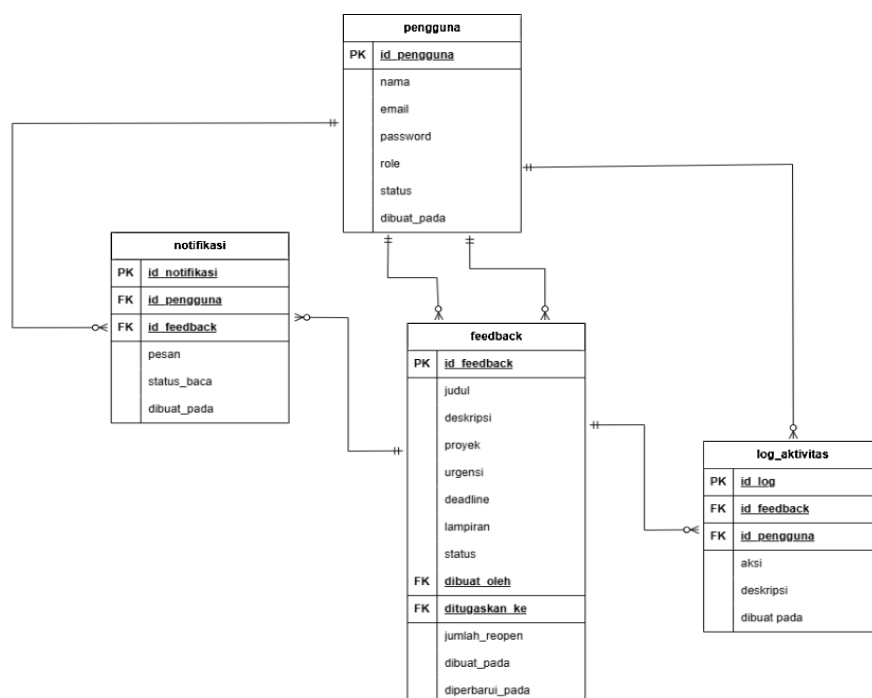
sistem mengirimkan notifikasi kepada *staff* untuk melakukan pengecekan hasil pengerjaan *feedback*. *Staff* kemudian melakukan pengecekan terhadap hasil *feedback* yang telah dikerjakan. Jika *feedback* telah sesuai, *staff* melakukan konfirmasi penyelesaian dan sistem akan memperbarui status *feedback* menjadi “Selesai”. Namun apabila *feedback* masih memerlukan perbaikan, *staff* dapat melakukan *reopen feedback* dan *assign* kembali kepada *programmer* untuk dilakukan perbaikan ulang hingga *feedback* dinyatakan selesai.



Gambar 4. Activity Diagram

3.3.3 Entity Relationship Diagram (ERD)

Struktur data dan hubungan antar entitas dalam sistem divisualisasikan menggunakan *Entity Relationship Diagram* (ERD). Entitas yang terlibat dalam pemodelan ini mencakup pengguna, *feedback*, notifikasi, serta *log aktivitas*, yang keseluruhannya saling terhubung untuk mendukung alur pengelolaan *feedback client*. Rancangan ERD yang diusulkan dapat dilihat pada Gambar 5.



Gambar 5. Entity Relationship Diagram

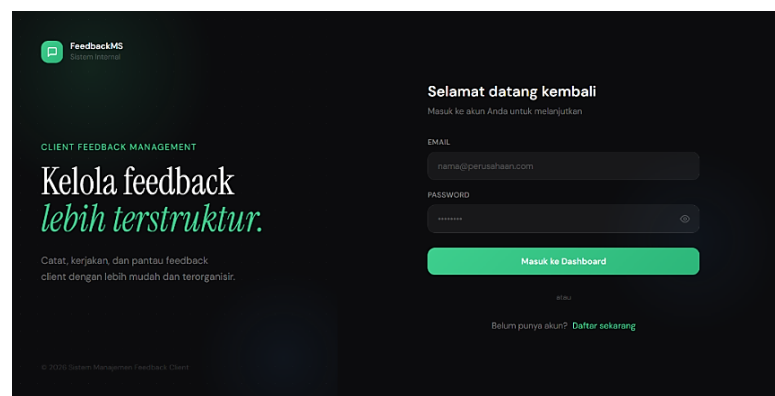
Gambar 5 menunjukkan *Entity Relationship Diagram* (ERD) yang digunakan sebagai dasar perancangan basis data sistem. ERD terdiri atas empat entitas utama, yaitu pengguna sebagai penyimpan data akun, *feedback* sebagai penyimpan data laporan, notifikasi sebagai media penyampaian informasi kepada pengguna, dan *log_aktivitas* sebagai pencatat seluruh aktivitas yang terjadi pada sistem. Relasi antarentitas tersebut mendukung proses pengelolaan *feedback* secara terstruktur dan terdokumentasi.

3.4 Implementasi Sistem

Tahap implementasi merupakan proses transformasi hasil analisis dan perancangan menjadi sebuah aplikasi web yang siap dioperasikan. Pada fase ini, seluruh spesifikasi kebutuhan yang telah dirumuskan sebelumnya diwujudkan dalam bentuk fitur-fitur fungsional yang dapat diakses oleh setiap pengguna sesuai dengan peran dan hak aksesnya masing-masing. Penggunaan platform berbasis web memungkinkan pengelolaan data berlangsung lebih efisien dan dapat diakses kapan saja melalui koneksi internet (Firmansah dkk., 2023). Berikut ini diuraikan tampilan antarmuka beserta fungsi dari masing-masing halaman yang tersedia dalam sistem:

a. Halaman Login

Halaman login merupakan halaman awal yang digunakan oleh pengguna untuk mengakses sistem. Pengguna diminta memasukkan email dan password yang telah terdaftar sesuai dengan peran masing-masing agar dapat masuk ke dalam sistem. Tampilan halaman login dapat dilihat pada Gambar 6.

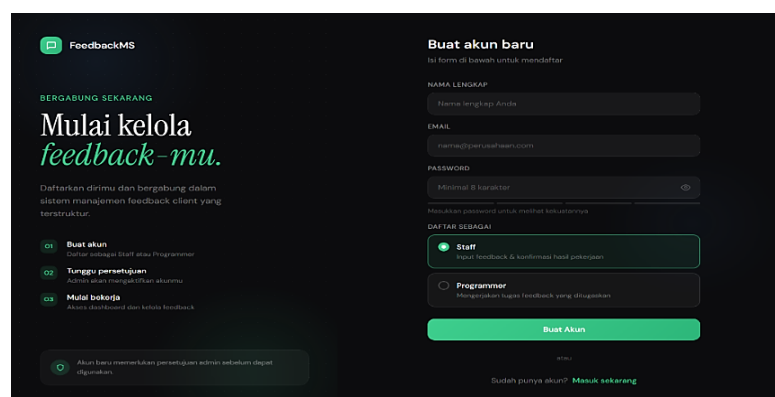


Gambar 6. Halaman Login

Halaman login menampilkan formulir autentikasi yang terdiri atas kolom email, password, serta tombol masuk ke sistem. Setelah proses autentikasi berhasil, pengguna akan diarahkan ke dashboard sesuai dengan hak akses yang dimilikinya.

b. Halaman Registrasi

Halaman registrasi merupakan halaman yang digunakan oleh calon pengguna untuk melakukan pendaftaran akun sebelum dapat mengakses sistem. Pada halaman ini pengguna diminta mengisi data diri serta memilih peran sesuai dengan tugasnya di dalam sistem. Tampilan halaman registrasi dapat dilihat pada Gambar 7.

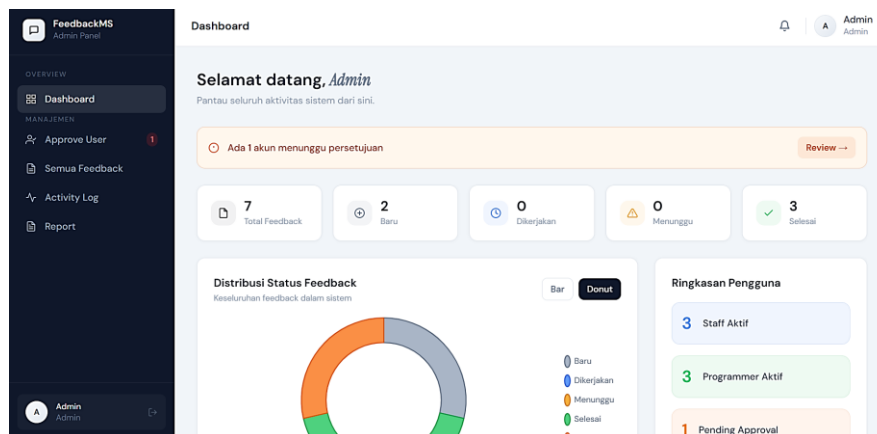


Gambar 7. Halaman Registrasi

Halaman registrasi menyediakan formulir yang berisi nama lengkap, email, password, dan pilihan role sebagai staff atau programmer. Akun yang telah didaftarkan selanjutnya menunggu proses persetujuan dari admin sebelum dapat digunakan untuk mengakses sistem.

c. Dashboard Admin

Dashboard admin merupakan halaman utama yang ditampilkan setelah admin berhasil melakukan login. Halaman ini menyajikan ringkasan informasi sistem serta menyediakan akses ke seluruh fitur yang dapat dikelola oleh admin. Tampilan dashboard admin dapat dilihat pada Gambar 8.

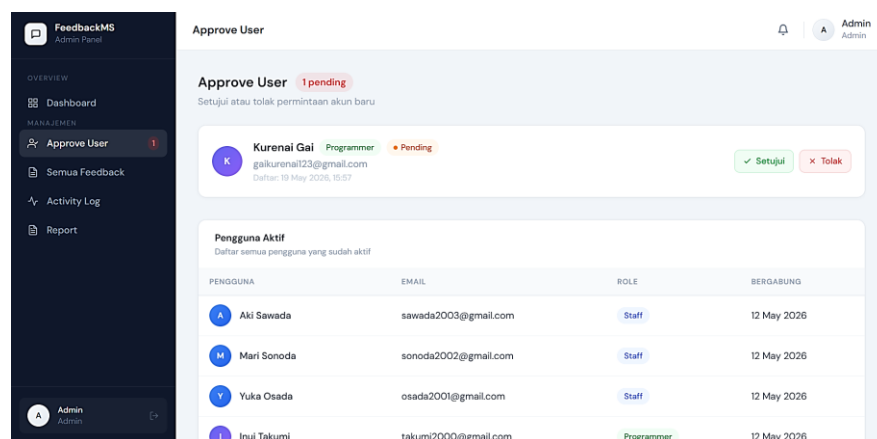


Gambar 8. Dashboard Admin

Dashboard admin menampilkan informasi jumlah feedback berdasarkan status, jumlah pengguna, grafik statistik, notifikasi, serta menu navigasi untuk mengelola pengguna, feedback, laporan, dan activity log sehingga memudahkan proses monitoring sistem.

d. *Approve User (Admin)*

Halaman approve user digunakan oleh admin untuk melakukan proses persetujuan terhadap akun yang telah didaftarkan oleh pengguna. Tampilan halaman approve user dapat dilihat pada Gambar 9.

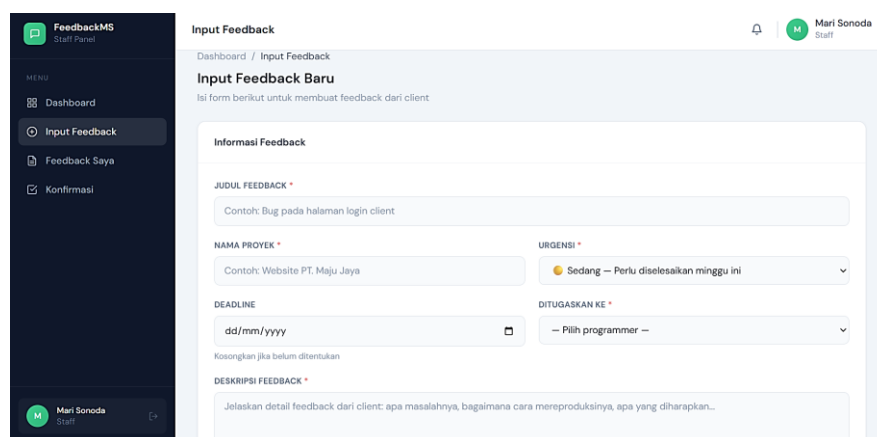


Gambar 9. Halaman Approve User

Halaman ini menampilkan daftar pengguna yang menunggu persetujuan beserta informasi nama, email, role, dan tanggal pendaftaran. Admin dapat menyetujui maupun menolak akun sehingga hanya pengguna yang telah disetujui yang dapat menggunakan sistem.

e. *Halaman Input Feedback (Staff)*

Halaman input feedback digunakan oleh staff untuk mencatat feedback yang diterima dari client ke dalam sistem. Seluruh informasi yang diperlukan dalam proses penanganan feedback diisikan melalui halaman ini. Tampilan halaman input feedback dapat dilihat pada Gambar 10.

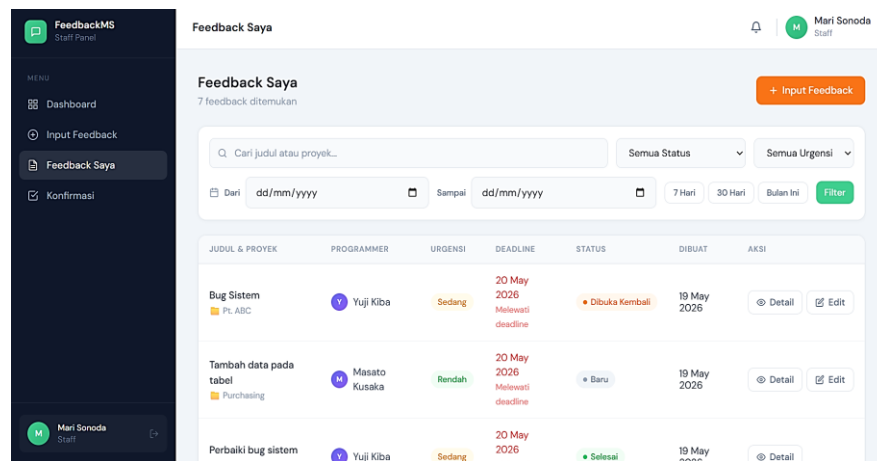


Gambar 10. Halaman Input Feedback

Gambar 10 merupakan halaman yang menyediakan formulir yang memuat data judul feedback, deskripsi, proyek terkait, tingkat urgensi, deadline, serta programmer yang akan ditugaskan. Setelah data disimpan, sistem akan mengirimkan notifikasi kepada programmer yang dipilih.

f. Halaman Daftar *Feedback (Staff)*

Halaman daftar feedback digunakan oleh staff untuk melihat seluruh feedback yang telah dicatat ke dalam sistem beserta perkembangan proses penanganannya. Tampilan halaman daftar feedback dapat dilihat pada Gambar 11.

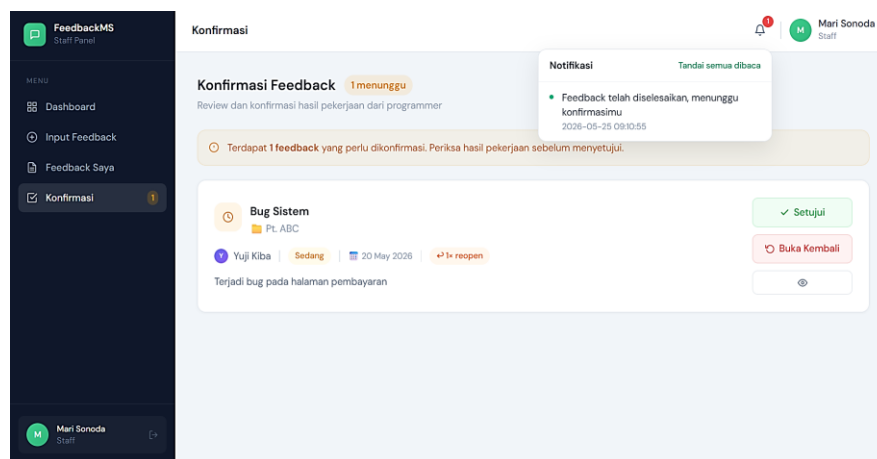


Gambar 11. Halaman Daftar *Feedback*

Halaman ini menyajikan daftar feedback dalam bentuk tabel yang berisi informasi judul, proyek, programmer yang ditugaskan, status penanganan, dan tanggal pembuatan. Staff juga dapat melakukan pencarian, penyaringan data, serta melihat detail feedback.

g. Halaman Konfirmasi Penanganan *Feedback (Staff)*

Halaman konfirmasi penanganan feedback digunakan oleh staff untuk melakukan verifikasi terhadap hasil pengerjaan feedback yang telah diselesaikan oleh programmer. Tampilan halaman konfirmasi penanganan feedback dapat dilihat pada Gambar 12.



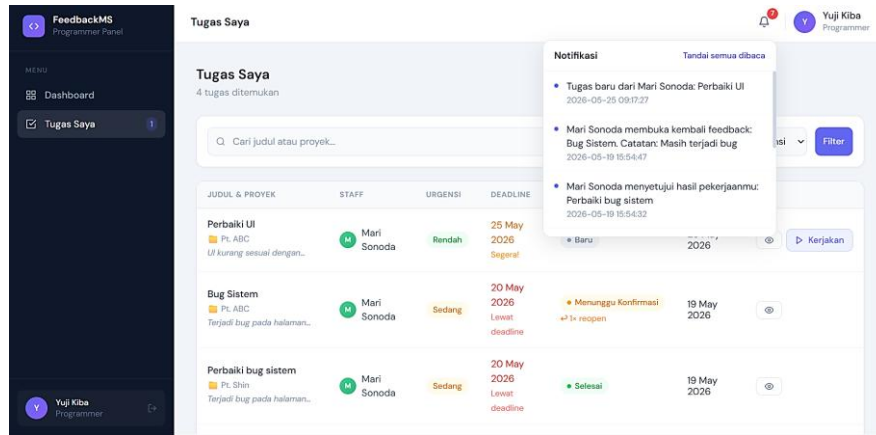
Gambar 12. Konfirmasi Penanganan *Feedback*

Gambar 12 menampilkan detail feedback beserta hasil penyelesaian yang diberikan oleh programmer. Staff dapat memberikan konfirmasi apabila hasil telah sesuai atau melakukan *reopen feedback* disertai catatan apabila masih diperlukan perbaikan.

h. Halaman Daftar Tugas (*Programmer*)

Halaman daftar tugas digunakan oleh programmer untuk melihat seluruh feedback yang menjadi tanggung jawabnya. Melalui halaman ini programmer dapat memantau daftar pekerjaan yang harus diselesaikan. Tampilan halaman daftar tugas dapat dilihat pada Gambar 13.

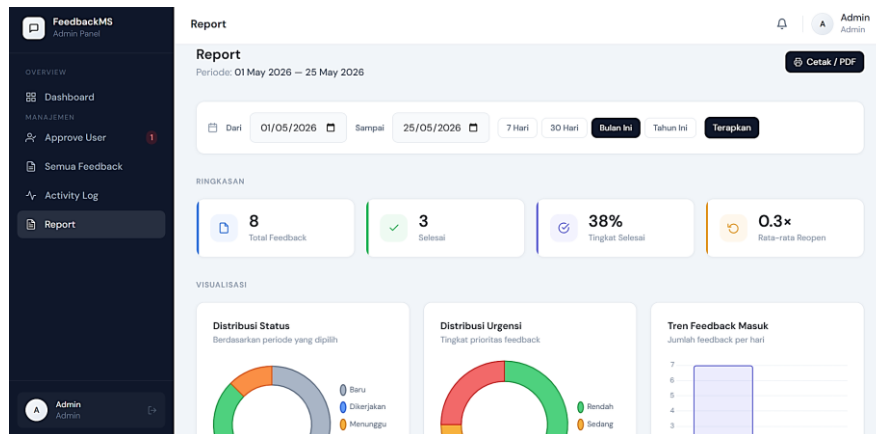
Gambar 13 menampilkan daftar tugas dalam bentuk tabel yang berisi informasi judul feedback, proyek, tingkat urgensi, deadline, serta status penanganan. Programmer dapat membuka detail tugas dan memperbarui status pengerjaan sesuai perkembangan pekerjaan.



Gambar 13. Halaman Daftar Tugas

i. Halaman *Report* (Admin)

Halaman report digunakan oleh admin untuk melihat laporan hasil pengelolaan feedback secara keseluruhan. Informasi yang disajikan membantu admin dalam memantau kinerja penanganan feedback pada periode tertentu. Tampilan halaman report dapat dilihat pada Gambar 14.



Gambar 14. Halaman *Report*

Halaman report menyajikan grafik dan tabel rekapitulasi feedback berdasarkan status, proyek, maupun programmer. Admin juga dapat melakukan penyaringan data berdasarkan rentang tanggal sehingga proses monitoring dan evaluasi menjadi lebih mudah.

3.5 Pengujian Sistem

Untuk memvalidasi bahwa seluruh fungsi sistem berjalan sebagaimana mestinya, pengujian dilaksanakan menggunakan metode *Blackbox Testing*. Pendekatan ini menitikberatkan pada evaluasi fungsional sistem berdasarkan data masukan dan keluaran yang dihasilkan, tanpa mempertimbangkan struktur internal kode program, sehingga efektif digunakan untuk memverifikasi kesesuaian perilaku sistem dengan kebutuhan pengguna yang telah ditetapkan. Rangkuman hasil pengujian disajikan secara lengkap pada Tabel 2 berikut:

Tabel 2. Pengujian Sistem menggunakan *BlackBox Testing*

Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
Login dengan data yang benar	Sistem berhasil masuk dan mengarahkan ke <i>dashboard</i> sesuai <i>role</i>	Sistem berhasil masuk dan menampilkan <i>dashboard</i> sesuai <i>role</i>	Berhasil
Login dengan data yang salah	Sistem menampilkan pesan “ <i>Email</i> dan <i>Password</i> salah” dan tidak dapat masuk	Sistem menampilkan pesan kesalahan sesuai	Berhasil
Login dengan akun berstatus pending	Sistem menampilkan pesan bahwa akun belum disetujui <i>Admin</i>	Sistem menampilkan pesan akun pending	Berhasil
Registrasi akun baru dengan data lengkap	Akun berhasil dibuat dengan status pending dan notifikasi terkirim ke <i>Admin</i>	Akun berhasil dibuat dan <i>Admin</i> menerima notifikasi pendaftaran	Berhasil
<i>Admin</i> menyetujui akun pengguna	<i>Status</i> akun akan berubah menjadi aktif dan pengguna dapat <i>login</i> ke sistem	Status berubah aktif, pengguna dapat masuk ke sistem	Berhasil



Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
<i>Admin</i> melihat laporan dengan filter periode	Sistem menampilkan laporan, grafik distribusi status, dan rekap performa <i>programmer</i> sesuai periode	Laporan tampil lengkap sesuai periode yang dipilih	Berhasil
<i>Staff</i> menginput <i>feedback</i> dengan data lengkap	<i>Feedback</i> berhasil tersimpan dengan status Baru dan notifikasi terkirim ke <i>Programmer</i>	<i>Feedback</i> tersimpan dan <i>Programmer</i> menerima notifikasi tugas baru	Berhasil
<i>Staff</i> menginput <i>feedback</i> tanpa mengisi <i>field</i> wajib	Sistem menampilkan pesan validasi dan <i>feedback</i> tidak tersimpan	Sistem menampilkan pesan validasi sesuai	Berhasil
<i>Staff</i> mengkonfirmasi penyelesaian <i>feedback</i>	Status <i>feedback</i> berubah menjadi Selesai dan notifikasi terkirim ke <i>Programmer</i>	Status berubah Selesai, <i>Programmer</i> menerima notifikasi	Berhasil
<i>Staff</i> melakukan <i>reopen feedback</i> disertai catatan	Status berubah menjadi Dibuka Kembali, catatan tersimpan, dan notifikasi beserta catatan terkirim ke <i>Programmer</i>	Status berubah dan <i>Programmer</i> menerima notifikasi dengan catatan	Berhasil
<i>Programmer</i> melihat daftar tugas	Sistem Menampilkan seluruh <i>feedback</i> yang ditugaskan beserta status dan tingkat urgensi	Seluruh tugas tampil dengan status dan urgensi yang sesuai	Berhasil
<i>Programmer</i> mengubah status <i>feedback</i> menjadi Dikerjakan	Status berubah menjadi Dikerjakan dan notifikasi terkirim ke <i>Staff</i>	Status berubah dan <i>Staff</i> menerima notifikasi	Berhasil
<i>Programmer</i> menandai <i>feedback</i> selesai dikerjakan	Status berubah menjadi Menunggu Konfirmasi dan notifikasi terkirim ke <i>Staff</i>	Status berubah dan <i>Staff</i> menerima notifikasi untuk melakukan konfirmasi	Berhasil
<i>Programmer</i> melihat catatan <i>reopen</i> dari <i>staff</i>	Sistem menampilkan catatan yang ditulis <i>Staff</i> saat melakukan <i>Reopen</i>	Catatan <i>Staff</i> tampil pada bagian detail <i>feedback</i>	Berhasil
Notifikasi muncul secara otomatis tanpa <i>refresh</i>	Notifikasi muncul secara <i>real-time</i> di panel pengguna yang bersangkutan tanpa perlu <i>reload</i> halaman	Notifikasi muncul otomatis sesuai perubahan yang terjadi	Berhasil

Berdasarkan Tabel 2, seluruh skenario pengujian memperoleh hasil Berhasil sesuai dengan keluaran yang diharapkan. Hasil tersebut menunjukkan bahwa setiap fungsi utama pada sistem, mulai dari proses autentikasi, registrasi pengguna, pengelolaan *feedback*, penugasan *programmer*, pembaruan status, hingga penyajian laporan, telah berjalan sesuai kebutuhan fungsional yang dirancang.

3.6 Pembahasan

Berdasarkan hasil implementasi dan pengujian sistem menggunakan *Blackbox Testing*, seluruh fungsi berjalan sesuai dengan hasil yang diharapkan. Hasil ini menunjukkan bahwa sistem informasi penanganan *feedback client* yang dikembangkan mampu menjawab permasalahan yang teridentifikasi melalui analisis PIECES, yang dimana aspek *performance*, *information*, *economy*, *control*, *efficiency*, dan *service* yang sebelumnya terganggu akibat proses manual kini dapat ditangani melalui sistem yang terstruktur dan terpusat. Temuan ini sejalan dengan penelitian (Susanto & Mulyati, 2023) yang mengembangkan sistem informasi *helpdesk* pada Diskominfo SP Kabupaten Banyuasin dan membuktikan bahwa sistem berbasis web mampu mendukung proses pencatatan keluhan, pengelolaan informasi, serta dokumentasi penanganan masalah secara terpusat sehingga pelayanan menjadi lebih terukur. Penelitian (Tarigan dkk., 2022) pada PT. Indonesia Nippon Seiki juga menunjukkan bahwa sistem *helpdesk ticketing* berbasis web yang memisahkan peran *user*, teknisi, dan pimpinan terbukti meningkatkan kualitas pelayanan melalui pemantauan status penyelesaian secara *real-time*. Senada dengan itu, (Rosmalina dkk., 2025) membuktikan bahwa sistem informasi keluhan berbasis web mampu meningkatkan transparansi dan kepuasan pengguna melalui fitur pelacakan status secara *real-time*, sementara (Sofyan dkk., 2024) menunjukkan bahwa sistem *ticketing* layanan berbasis web yang pengujiannya dilakukan dengan metode *Black Box Testing* menunjukkan bahwa seluruh fungsi sistem beroperasi sesuai dengan spesifikasi yang telah ditetapkan dan mampu meningkatkan efisiensi penanganan laporan secara signifikan. Perbedaan mendasar penelitian ini dibandingkan keempat penelitian tersebut terletak pada konteks dan alur kerja yang dikembangkan. Sistem yang dibangun secara khusus dirancang untuk lingkungan *software house* dengan mengintegrasikan tiga peran utama, yaitu *admin*, *programmer*, dan *staff* dalam satu alur penanganan *feedback* yang runtut. Sistem juga menyediakan mekanisme *reopen feedback* yang memungkinkan *staff* mengembalikan *feedback* kepada *programmer* disertai catatan apabila hasil penanganan belum sesuai kebutuhan, sehingga proses tindak lanjut dapat berjalan lebih terkontrol dan sesuai dengan alur kerja nyata di BIIS Corp.



4. KESIMPULAN

Melalui penelitian ini, telah dihasilkan sebuah sistem informasi berbasis web untuk penanganan *feedback client* di lingkungan BIIS Corp. yang dikembangkan dengan menerapkan pendekatan metode *Waterfall*. Sebelum pengembangan sistem dilakukan, analisis menggunakan *framework* PIECES menunjukkan adanya permasalahan pada aspek *performance*, *information*, *economy*, *control*, *efficiency*, dan *service* yang disebabkan oleh penggunaan *WhatsApp* dan *spreadsheet* sebagai media utama pengelolaan *feedback*. Untuk mengatasi permasalahan tersebut, dikembangkan sistem yang mengintegrasikan peran *admin*, *staff*, dan *programmer* dalam satu platform terpusat. Sistem menyediakan fitur pencatatan *feedback*, penugasan *programmer*, monitoring status penanganan, konfirmasi penyelesaian, mekanisme *reopen feedback*, notifikasi *real-time*, *activity log*, dan laporan. Kontribusi penelitian ini terletak pada pengembangan sistem yang dirancang sesuai alur operasional software house dengan menyediakan mekanisme penugasan programmer, monitoring status, *reopen feedback*, notifikasi *real-time*, dan *activity log* dalam satu platform terintegrasi sehingga proses pengelolaan *feedback* menjadi lebih terstruktur dibandingkan proses manual sebelumnya. Berdasarkan hasil pengujian menggunakan metode *Blackbox Testing* terhadap 15 skenario pengujian, seluruh fungsi sistem berjalan sesuai dengan hasil yang diharapkan dengan tingkat keberhasilan sebesar 100%, sehingga sistem telah memenuhi kebutuhan fungsional yang ditetapkan pada tahap analisis. Meskipun demikian, penelitian ini masih memiliki keterbatasan karena sistem belum diterapkan pada lingkungan operasional BIIS Corp dan pengujian yang dilakukan masih terbatas pada aspek fungsional. Oleh karena itu, penelitian selanjutnya dapat difokuskan pada proses *deployment* sistem pada lingkungan operasional perusahaan serta evaluasi tingkat penerimaan pengguna melalui penyebaran kuesioner untuk memperoleh umpan balik mengenai kemudahan penggunaan dan manfaat sistem dalam mendukung pengelolaan *feedback client*.

REFERENCES

- Anis, Y., Wahyudi, E. N., & Kurniawan, H. C. (2024). Metode Waterfall dalam Pengembangan Sistem Inventaris Guna Meningkatkan Efisiensi Manajemen Stok Barang. *Jurnal Teknologi Dan Sistem Informasi Bisnis*, 6(2), 322–328. <https://doi.org/10.47233/jteksis.v6i2.1350>
- Damayanti, E., Sanjaya, W., & Wulandari, F. T. (2023). Sistem Pengaduan Masyarakat Berbasis Web menggunakan Metode Waterfall. *JITU: Journal Informatic Technology And Communication*, 7(2), 147–153. <https://doi.org/10.36596/jitu.v7i2.818>
- Fachri, B., Rizal, C., & Supiyandi. (2024). Penerapan Metode Waterfall Dalam Perancangan Sistem Informasi Merdeka Belajar Kampus Merdeka Berbasis Web. *Jurnal Komputer Teknologi Informasi Sistem Komputer (JUKTISI)*, 2(3), 591–597. <https://doi.org/https://doi.org/10.62712/juktisi.v2i3.147>
- Firmansah, A., Mahendra, D., Minardi, J., Saputro, H., & Sabilla, A. D. (2023). Implementasi Sistem Informasi Berbasis Web Untuk Pengelolaan Jasa Penyewaan Internet (Studi Kasus: Bumdes Desa Kerso). *Journal of Information System and Computer*, 3(2), 1. <https://doi.org/https://doi.org/10.34001/jister.v3i2.649>
- Fuaddin, A. F., Pratama, F. T. Y. P., Putra, F. A. M., & Setiawan, I. (2025). Analisis Tingkat Kepuasan Pengguna Sistem Informasi Perpustakaan Dengan Pendekatan Pieces Framework. *Merkurius: Jurnal Riset Sistem Informasi dan Teknik Informatika*, 3(1), 272–282. <https://doi.org/10.61132/merkurius.v3i1.665>
- Hendry, Supiyandi, Rizal, C., Eka, M., & Zulham. (2025). Implementasi Sistem Informasi Pengaduan Masyarakat Berbasis Web dengan Automatic Ticketing Workflow. *Jurnal Komputer Teknologi Informasi Sistem Informasi (JUKTISI)*, 4(2), 1411–1416. <https://doi.org/10.62712/juktisi.v4i2.694>
- Irawan, D., Darmawan, E. Y., Zebua, E. E., & Haryono, W. (2024). Perancangan Sistem Informasi Proyek Berbasis Web Untuk Meningkatkan Kinerja Antar Divisi. *Jurnal Komputer Antartika*, 2(4), 136–146. <https://doi.org/10.70052/jka.v2i4.629>
- Maulana, A., & Budiyantra, A. (2025). Penerapan Analisa PIECES dan Rancang Bangun Sistem Informasi Inventaris Kantor dengan Pemodelan UML Berbasis Website. *Jurnal ilmiah Sistem Informasi dan Ilmu Komputer*, 5(3), 430–445. <https://doi.org/10.55606/juisik.v5i3.1736>
- Mufarikuddin, A., & Surianto, M. A. (2022). Analisis Penanganan Keluhan Pelanggan Pada Pt. Pos Indonesia Surabaya 60000. *Jurnal Ekonomika*45, 10(1), 118–123. <https://jurnaluniv45sby.ac.id/index.php/ekonomika/article/view/551>
- Muhyiddin, A. H., Kharisma, A. P., & Ananta, M. T. (2023). Pengembangan Customer Relationship Management (CRM) Sistem Umpan Balik Pelanggan (Customer Feedback System) pada Perusahaan XYZ. *Jurnal Pengembangan Teknologi Informasidan Ilmu Komputer*, 7(2), 663–670. <https://doi.org/https://doi.org/10.30640/ekonomika45.v10i1.551>
- Pratomo, A. B. (2023). Pengembangan Website Manajemen Tugas Berbasis Web Untuk Pengukuran Kinerja Sumber Daya Manusia. *Teknimedia*, 4(2), 139–143. <https://doi.org/https://doi.org/10.46764/teknimedia.v4i2.113>
- Rosmalina, Nistrina, K., Rusdianto, D., & Mujadi, Y. (2025). Pengembangan Sistem Informasi Keluhan Pelanggan Berbasis Laravel Untuk Peningkatan Layanan Jasa Nugraha Ekakurir (JNE). *Jurnal Pendidikan dan Teknologi Indonesia*, 5(1), 131–142. <https://doi.org/10.52436/1.jpti.601>
- Silitonga, N. A., Samartha, L. H. D., Wiguna, T. A., Baidawi, M., & Saifudin, A. (2023). OKTAL: Jurnal Ilmu Komputer dan Science Pengujian Black Box Testing Pada Aplikasi Absensi Berbasis Web Di Sekolah SDN Duri



- Kepa 01. *OKTAL: Jurnal Ilmu Komputer dan Science*, 2(5), 1383–1395. <https://journal.mediapublikasi.id/index.php/oktal/article/download/1270/1338>
- Sofyan, A., Prastyo, A. D., Saputra, A., & Djutalov, R. (2024). Pengembangan Sistem Ticketing Layanan Menggunakan Metode Agile Berbasis Website Pada PT. Qonita Teknologi Indonesia. *Jurnal Komputer Antartika*, 2(4), 164–171. <https://doi.org/10.70052/jka.v2i4.646>
- Susanto, S., & Mulyati. (2023). Sistem Informasi Helpdesk Dalam Tata Kelola Teknologi Informasi Pada Diskominfo dan SP. *JuTI "Jurnal Teknologi Informasi,"* 2(1), 63. <https://doi.org/10.26798/juti.v2i1.961>
- Tarigan, R., Kusosi, I., & Usri, A. (2022). Perancangan Aplikasi Helpdesk Ticketing System Pada PT. Indonesia Nippon Seiki. *Jurnal Sisfokom (Sistem Informasi dan Komputer)*, 11(1), 9–18. <https://doi.org/10.32736/sisfokom.v11i1.1271>
- Zen, M., Irwan, Hafni, & Ananda, M. D. P. (2024). Implementasi dan Pengujian Menggunakan Metode BlackBox Testing Pada Sistem Informasi Tracer Study. *Bulletin of Computer Science Research*, 4(4), 327–340. <https://doi.org/10.47065/bulletincsr.v4i4.359>
- Zunindar, & Wiyono, N. (2023). Analisa Dan Perancangan Sistem Informasi Pelayanan Pengaduan Masyarakat Di Kecamatan Rajeg Berbasis Web. *Jurnal Manajemen Informatika Jayakarta*, 3(1), 87–97. <https://doi.org/https://doi.org/10.52362/jmijayakarta.v3i1.1018>