IMPLEMENTASI PREPARED STATEMENT PADA FORM INPUT UNTUK MENCEGAH SQL INJECTION PADA APLIKASI WEB PEMINJAMAN FASILITAS

Wulan Febri Enjelina, Agus Tedyyana
Keamanan Sistem Informasi, Politeknik Negeri Bengkalis
Jalan Bathin Alam Desa Sungai Alam, Kec. Bengkalis, Kab. Bengkalis, Indonesia
wulanfebrienjelina@gmail.com

ABSTRAK

Penerapan aplikasi berbasis web dalam pengelolaan peminjaman fasilitas kampus menjadi solusi strategis untuk meningkatkan efisiensi administrasi dan ketertiban data. Namun, sistem web yang dikembangkan tanpa pengamanan yang memadai berisiko mengalami serangan keamanan, khususnya SQL Injection. Di Politeknik Negeri Bengkalis, proses peminjaman fasilitas masih dilakukan secara manual dan semi-digital, sehingga menimbulkan permasalahan berupa konflik jadwal, ketidakkonsistenan data, serta lemahnya pengawasan keamanan sistem. Penelitian ini bertujuan untuk merancang dan mengimplementasikan aplikasi web peminjaman fasilitas kampus yang aman dengan menerapkan Prepared Statement pada seluruh form input yang terhubung dengan basis data. Metode pengembangan yang digunakan adalah Waterfall, yang meliputi tahap analisis kebutuhan, perancangan sistem, implementasi, pengujian, dan pemeliharaan. Prepared Statement diterapkan untuk memisahkan struktur perintah SQL dari data input pengguna sehingga mencegah input dieksekusi sebagai perintah berbahaya. Pengujian keamanan dilakukan menggunakan SQLMap untuk mengidentifikasi potensi kerentanan SQL Injection pada parameter input sistem. Hasil pengujian menunjukkan bahwa seluruh fitur aplikasi tidak memiliki celah SQL Injection dan mampu menjaga integritas serta kerahasiaan data. Dengan demikian, aplikasi yang dikembangkan tidak hanya meningkatkan efektivitas pengelolaan peminjaman fasilitas kampus, tetapi juga memperkuat aspek keamanan sistem informasi berbasis web secara menyeluruh.

Kata kunci : *Prepared Statement, Sql Injection, Peminjaman Fasilitas*

1. PENDAHULUAN

Pengembangan teknologi informasi telah mendorong institusi pendidikan tinggi untuk menggunakan sistem informasi berbasis web untuk membantu administrasi dan penyediaan layanan akademik. Dibandingkan dengan metode manual, aplikasi berbasis web dinilai dapat meningkatkan kecepatan, efisiensi, dan akurasi pengelolaan data**.**

Peminjaman fasilitas kampus, termasuk peralatan pendukung kegiatan akademik, laboratorium, dan ruang kelas, adalah salah satu layanan administrasi yang membutuhkan sistem terintegrasi. Pengelolaan peminjaman yang masih dilakukan secara manual atau semi-digital dapat menyebabkan beberapa masalah, seperti konflik jadwal, data yang tidak konsisten, kesulitan untuk melacak riwayat peminjaman, dan rendahnya efisiensi pengawasan penggunaan fasilitas[1].

Sebaliknya, penggunaan aplikasi berbasis web juga membawa risiko keamanan yang signifikan. Aplikasi yang menerima input pengguna, terutama pada fitur login dan formulir pengajuan, sangat rentan terhadap serangan SQL Injection jika tidak dilengkapi dengan mekanisme validasi dan pengamanan yang tepat. SQL Injection adalah teknik serangan yang memanfaatkan kelemahan pemrosesan input untuk menyisipkan perintah SQL berbahaya, memungkinkan penyerang untuk mengakses, mengubah, atau menghapus data secara tidak sah[2]. Berdasarkan laporan OWASP, serangan injection masih termasuk dalam ancaman paling kritis terhadap keamanan aplikasi web [3] Data nasional juga menunjukkan bahwa SQL Injection menjadi salah satu jenis serangan siber yang paling sering dilaporkan di Indonesia[4]

Proses peminjaman fasilitas di Politeknik Negeri Bengkalis belum sepenuhnya terintegrasi ke dalam sistem web yang aman dan terstandar. Kondisi ini menghambat pengelolaan fasilitas dan berpotensi menimbulkan risiko keamanan jika sistem dikembangkan tanpa perlindungan terhadap serangan SQL Injection. Salah satu metode yang terbukti efektif dalam mencegah serangan tersebut adalah Prepared Statement. Teknik ini bekerja dengan memisahkan struktur perintah SQL dari data input pengguna, sehingga input tidak dapat diproses sebagai bagian dari perintah SQL dan tidak berpotensi menimbulkan injeksi[5]

Berdasarkan masalah tersebut, penelitian ini bertujuan untuk merancang dan menerapkan aplikasi web untuk peminjaman fasilitas kampus. Aplikasi ini akan menggunakan Pernyataan yang Disiapkan pada semua form input yang terhubung ke basis data. Agar setiap tahapan pengembangan dapat berjalan secara sistematis dan terdokumentasi dengan baik, pengembangan sistem dilakukan menggunakan metode Waterfall. Metode ini diharapkan dapat meningkatkan efektivitas manajemen pinjaman fasilitas dan meningkatkan keamanan dan integritas data di kampus.

Proses peminjaman fasilitas kampus masih dilakukan secara manual dan sebagian digital, yang

menyebabkan banyak masalah. Beberapa di antaranya adalah konflik jadwal, pencatatan data yang tidak konsisten, dan kesulitan untuk melacak riwayat penggunaan fasilitas. Kondisi ini menyebabkan pengelolaan fasilitas kampus tidak efisien dan efektif. Selain itu, menggunakan aplikasi web sebagai alat digital menimbulkan ancaman keamanan, terutama untuk fitur yang menerima input pengguna. Penggunaan query database yang tidak aman dapat menyebabkan kerentanan terhadap serangan SQL Injection, yang dapat membocorkan, mengubah, atau menghapus data secara tidak sah. Hingga saat ini, sistem peminjaman fasilitas kampus belum sepenuhnya menerapkan mekanisme Pernyataan Siap. Akibatnya, aspek keamanan input masih menjadi masalah utama yang perlu ditangani**.**

Tujuan penelitian ini adalah untuk membuat aplikasi web yang terintegrasi untuk peminjaman di fasilitas kampus yang meningkatkan kemudahan, akurasi, dan efisiensi pengelolaan data peminjaman. Selain itu, sebagai upaya mencegah serangan SQL Injection, penelitian ini bertujuan untuk menerapkan *Prepared Statement* pada form input yang terhubung ke basis data. Tujuan tambahan adalah menguji tingkat keamanan aplikasi web yang dibuat untuk memastikan bahwa sistem tidak memiliki celah injeksi dan dapat mempertahankan integritas dan kerahasiaan data. Dengan demikian, diharapkan sistem yang dibuat dapat digunakan sebagai metode peminjaman fasilitas kampus yang aman dan andal.

Rencana penyelesaian masalah dilakukan melalui pengembangan sistem menggunakan metode Waterfall yang berjalan secara sistematis dan terstruktur. Tahap awal dimulai dengan analisis kebutuhan untuk mengidentifikasi permasalahan operasional serta titik input yang berpotensi menjadi celah keamanan. Selanjutnya, dilakukan perancangan sistem yang mencakup alur proses, desain basis data, dan perancangan antarmuka pengguna. Tahap implementasi dilakukan dengan mengembangkan aplikasi web berbasis PHP dan MySQL serta menerapkan *Prepared Statement* pada query yang melibatkan input pengguna. Setelah sistem diimplementasikan, dilakukan pengujian keamanan menggunakan SQLMap untuk memastikan tidak adanya kerentanan *SQL Injection*. Tahap akhir adalah evaluasi dan penyempurnaan sistem berdasarkan hasil pengujian agar aplikasi dapat digunakan secara optimal dan aman di lingkungan kampus.

2. TINJAUAN PUSTAKA

Bagian ini membahas konsep dasar dan penelitian terdahulu tentang aplikasi web peminjaman fasilitas, dengan keamanan *Prepared Statement*. Tinjauan pustaka digunakan sebagai landasan teoritis untuk mendukung perancangan dan analisis sistem yang dikembangkan.

2.1. Aplikasi Berbasis Web

Aplikasi berbasis web banyak digunakan dalam lingkungan akademik karena kemudahan akses, fleksibilitas perangkat, serta efisiensi dalam pengelolaan data. Penerapan sistem informasi berbasis web terbukti mampu meningkatkan akurasi pencatatan dan mempercepat layanan administrasi dibandingkan metode manual. Dalam konteks kampus, sistem peminjaman fasilitas berbasis web membantu mengelola data peminjaman, ketersediaan fasilitas, serta riwayat penggunaan secara terintegrasi, sehingga mengurangi konflik jadwal dan kesalahan pencatatan.[6]

2.2. Keamanan Aplikasi Web

Meskipun aplikasi web mudah digunakan, mereka memiliki tingkat ancaman keamanan yang tinggi, terutama pada fitur yang menerima input pengguna. Serangan injection, yang masih menempati kategori risiko tinggi pada aplikasi web kontemporer menurut laporan OWASP Top 10 tahun 2021, adalah salah satu ancaman paling berbahaya.[3] Ancaman ini muncul akibat lemahnya validasi input dan pengelolaan query database.

Di Indonesia, serangan SQL Injection masih sering ditemukan pada berbagai aplikasi web. Laporan keamanan siber menunjukkan bahwa SQL Injection menjadi salah satu aduan tertinggi dalam insiden keamanan aplikasi web dan terus mengalami peningkatan dalam beberapa tahun terakhir[4]

2.3. SQL Injection

Teknik serangan yang dikenal sebagai SQL Injection menggunakan celah dalam pemrosesan input pengguna untuk memasukkan perintah SQL berbahaya ke dalam query database. Serangan ini dapat menyebabkan data sensitif bocor, perubahan data tanpa izin, atau penghapusan data sepenuhnya.[2] Penelitian sebelumnya menunjukkan bahwa penggunaan query dinamis tanpa mekanisme pengamanan yang tepat menjadi penyebab utama terjadinya kerentanan SQL Injection pada aplikasi web[7]

2.4 *Prepared Statement*

Salah satu cara untuk mencegah serangan injeksi SQL adalah dengan membuat struktur perintah SQL terpisah dari data input pengguna. Dengan cara ini, input tidak dapat diproses sebagai bagian dari perintah SQL, sehingga serangan injeksi dapat dicegah.Studi menunjukkan bahwa menerapkan *prepared statement* secara teratur pada semua permintaan database dapat menghilangkan kerentanan SQL Injection dan meningkatkan keamanan aplikasi web.[8]

Penelitian terkait sistem peminjaman fasilitas kampus umumnya berfokus pada pengembangan sistem berbasis web dan peningkatan efisiensi operasional menggunakan metode Waterfall[9]. Namun, sebagian besar penelitian tersebut belum menempatkan aspek keamanan, khususnya

pencegahan SQL Injection, sebagai fokus utama. Di sisi lain, penelitian yang membahas SQL Injection lebih banyak menekankan analisis dan pengujian keamanan tanpa mengintegrasikannya langsung ke dalam sistem peminjaman kampus.[10]

2.5 Penelitian Terdahulu

Pengembangan sistem informasi berbasis web dalam lingkungan akademik telah banyak dikaji sebagai solusi untuk meningkatkan efisiensi layanan administrasi. Penelitian [11] mengembangkan sebuah sistem akreditasi program studi berbasis web untuk menggantikan proses pengelolaan dokumen yang sebelumnya dilakukan secara manual. Sistem tersebut dirancang menggunakan model pengembangan Waterfall dan dilengkapi dengan mekanisme pembagian peran pengguna, sehingga setiap pengguna hanya dapat mengakses fitur sesuai dengan tanggung jawabnya. Hasil pengujian menunjukkan bahwa penerapan sistem berbasis web mampu mempercepat proses administrasi, mengurangi beban kerja manual, serta meningkatkan efisiensi pengelolaan data akreditasi. Meskipun demikian, fokus utama penelitian ini lebih menekankan pada aspek fungsional dan manajemen dokumen, sementara pembahasan mengenai keamanan aplikasi web belum dibahas secara mendalam.

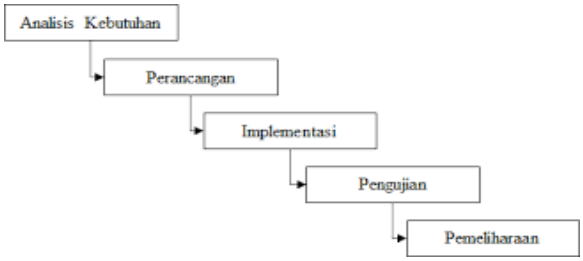
Dari sisi keamanan aplikasi web, penelitian [1] menerapkan Web Application Firewall (WAF) sebagai upaya perlindungan terhadap serangan eksternal pada sebuah website. Penelitian ini membuktikan bahwa WAF dapat mengurangi lalu lintas berbahaya dan meningkatkan keamanan sistem. Namun, pendekatan yang digunakan lebih berfokus pada lapisan jaringan dan belum secara khusus membahas perlindungan terhadap query database dari serangan SQL Injection.

Penelitian yang secara spesifik membahas SQL Injection, yang mengidentifikasi kerentanan aplikasi web melalui vulnerability assessment [2]. Hasilnya menunjukkan bahwa lemahnya validasi input dan penggunaan query dinamis menjadi penyebab utama munculnya celah SQL Injection. Temuan tersebut sejalan dengan penelitian [8], yang menegaskan bahwa SQL Injection merupakan salah satu serangan paling berisiko karena dapat memberikan akses tidak sah terhadap basis data.

Sebagai upaya pencegahan, beberapa penelitian merekomendasikan penggunaan *Prepared Statement*. menyatakan bahwa pemisahan antara struktur perintah SQL dan data input pengguna efektif dalam menekan potensi SQL Injection. Penelitian [12] juga menunjukkan bahwa penggunaan *Prepared Statement* melalui PHP Data Object (PDO) memberikan tingkat keamanan yang lebih tinggi dibandingkan query konvensional. Efektivitas metode ini akan optimal apabila diterapkan secara konsisten pada seluruh query yang berinteraksi dengan database[13].

Berdasarkan kajian terhadap penelitian terdahulu, dapat disimpulkan bahwa sebagian besar penelitian masih memisahkan fokus antara pengembangan sistem peminjaman berbasis web dan penerapan keamanan database. Oleh karena itu, penelitian ini mengintegrasikan kedua aspek tersebut dengan membangun aplikasi web peminjaman fasilitas kampus yang menerapkan *Prepared Statement* sebagai mekanisme keamanan utama serta melakukan pengujian menggunakan SQLMap untuk memastikan sistem aman dari serangan SQL Injection.

3. METODE PENELITIAN



Gambar 1. Metode *Waterfall*

Penelitian ini menggunakan metode pengembangan perangkat lunak *Waterfall* yang memiliki alur kerja terstruktur dan sesuai untuk sistem dengan kebutuhan yang telah didefinisikan sejak awal. Metode ini terdiri dari tahapan analisis kebutuhan, perancangan sistem, implementasi, pengujian, dan pemeliharaan. Pendekatan Waterfall dipilih agar setiap tahapan dapat dilakukan secara sistematis dan terdokumentasi dengan baik, yang memudahkan proses evaluasi dan pengujian.

3.1. Analisis Kebutuhan

Tahap analisis kebutuhan dilakukan untuk mengidentifikasi permasalahan pada proses peminjaman fasilitas kampus yang masih bersifat manual dan semi-digital. Pada tahap ini juga dianalisis kebutuhan fungsional sistem, seperti pengelolaan data fasilitas, peminjaman, pengembalian, dan manajemen pengguna, serta kebutuhan non-fungsional yang berfokus pada aspek keamanan, khususnya potensi kerentanan SQL Injection pada form input yang terhubung ke basis data.

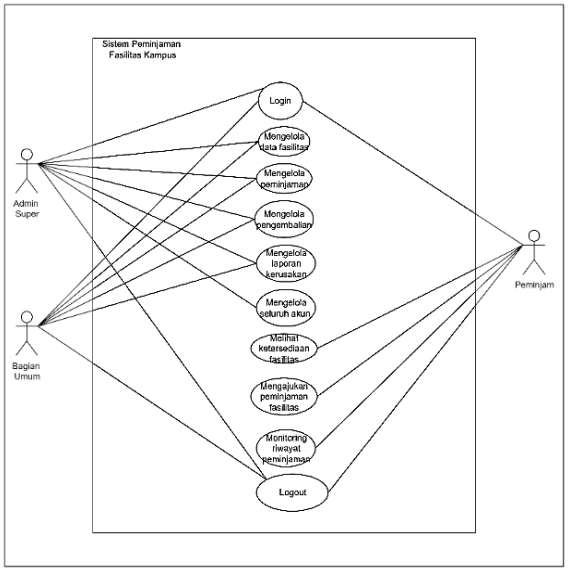
3.2. Perancangan Sistem

Tujuan dari tahap perancangan sistem adalah untuk mengubah kebutuhan pengguna menjadi rancangan teknis. Alur proses sistem, use case diagram, dan activity diagram, serta rancangan basis data relasional, semua merupakan bagian dari proses perancangan. Pada tahap ini, mekanisme keamanan juga dirancang dengan menerapkan *Prepared Statement* pada *query database* untuk membedakan perintah SQL dari data input pengguna.

3.3. *Use Case Diagram*

Use case diagram diatas digunakan untuk menunjukkan bagaimana pengguna berinteraksi dengan sistem peminjaman fasilitas kampus. Diagram ini menjelaskan siapa saja yang bisa menggunakan

sistem tersebut, fitur-fitur yang tersedia, serta bagaimana hubungan antara pengguna dengan berbagai fungsi dalam sistem. Terdapat tiga jenis pengguna utama dalam sistem ini, yaitu peminjam, bagian umum, dan super admin.

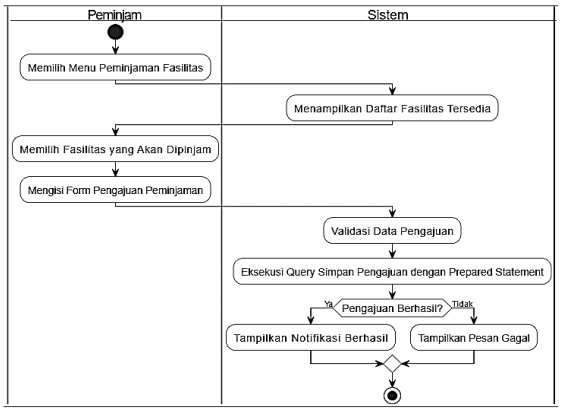


Gambar 2. Use case diagram

3.4. Activity Diagram

Activity Diagram adalah salah satu diagram dalam pemodelan UML (Unified Modeling Language), yang *digunakan* untuk menunjukkan alur aktivitas atau alur kerja yang terjadi di dalam sistem informasi. Diagram ini menunjukkan urutan langkah-langkah proses dari aktivitas awal hingga aktivitas akhir, yang mencakup keputusan atau percabangan yang mungkin terjadi

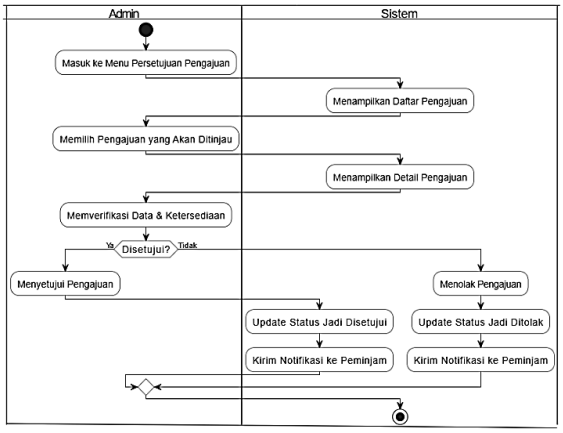
3.5. *Activity Diagram Pengajuan peminjaman fasilitas*



Gambar 3. Peminjaman Fasilitas

Activity diagram di atas menunjukkan alur kerja proses pengajuan peminjaman fasilitas kampus yang terjadi antara Peminjam dan Sistem. Proses dimulai dengan *Peminjam* memilih menu "fasilitas peminjaman", kemudian sistem menampilkan daftar fasilitas yang tersedia. Setelah itu, Peminjam memilih fasilitas mana yang ingin mereka pinjam dan mengisi formulir pengajuan peminjaman. Semua data ini kemudian diverifikasi oleh sistem untuk memastikan bahwa mereka benar dan lengkap. Untuk menghindari serangan SQL Injection, prepared statement digunakan untuk menyimpan data pengajuan yang valid ke dalam basis data. Setelah proses penyimpanan selesai, sistem menampilkan pesan keberhasilan atau pesan kegagalan jika terjadi kesalahan, dan proses pengajuan dinyatakan selesai.

3.6. *Activity Diagram Persetujuan Pengajuan oleh Admin*



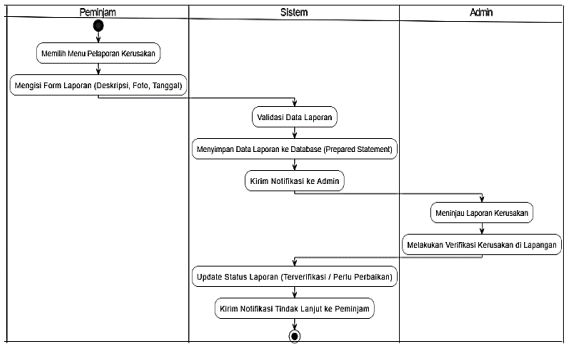
Gambar 4 .Persetujuan Pengajuan oleh Admin

Activity diagram diatas menggambarkan proses persetujuan pengajuan peminjaman fasilitas yang dilakukan oleh Admin. Proses diawali ketika admin masuk ke menu persetujuan pengajuan, kemudian sistem menampilkan daftar pengajuan yang masuk. Admin memilih salah satu pengajuan untuk ditinjau, dan sistem menampilkan detail pengajuan yang diajukan oleh peminjam. Selanjutnya, admin melakukan verifikasi data pengajuan serta ketersediaan fasilitas. Berdasarkan hasil verifikasi tersebut, admin dapat menyetujui atau menolak pengajuan. Jika pengajuan disetujui, sistem akan memperbarui status menjadi disetujui dan mengirimkan notifikasi kepada peminjam. Sebaliknya, jika pengajuan ditolak, sistem akan mengubah status menjadi ditolak dan tetap mengirimkan notifikasi kepada peminjam, kemudian proses persetujuan dinyatakan selesai.

3.7. *Activity Diagram Proses pelaporan kerusakan*

Activity diagram pada gambar 5 menjelaskan proses pelaporan kerusakan fasilitas yang melibatkan *Peminjam*, Sistem, dan Admin. Alur dimulai saat Peminjam mengakses menu pelaporan kerusakan dan mengisi formulir laporan yang memuat informasi deskripsi kerusakan, dokumentasi foto, serta waktu kejadian. Selanjutnya, sistem melakukan pemeriksaan kelengkapan data dan menyimpan laporan ke dalam basis data menggunakan *prepared statement* untuk

memastikan keamanan data.

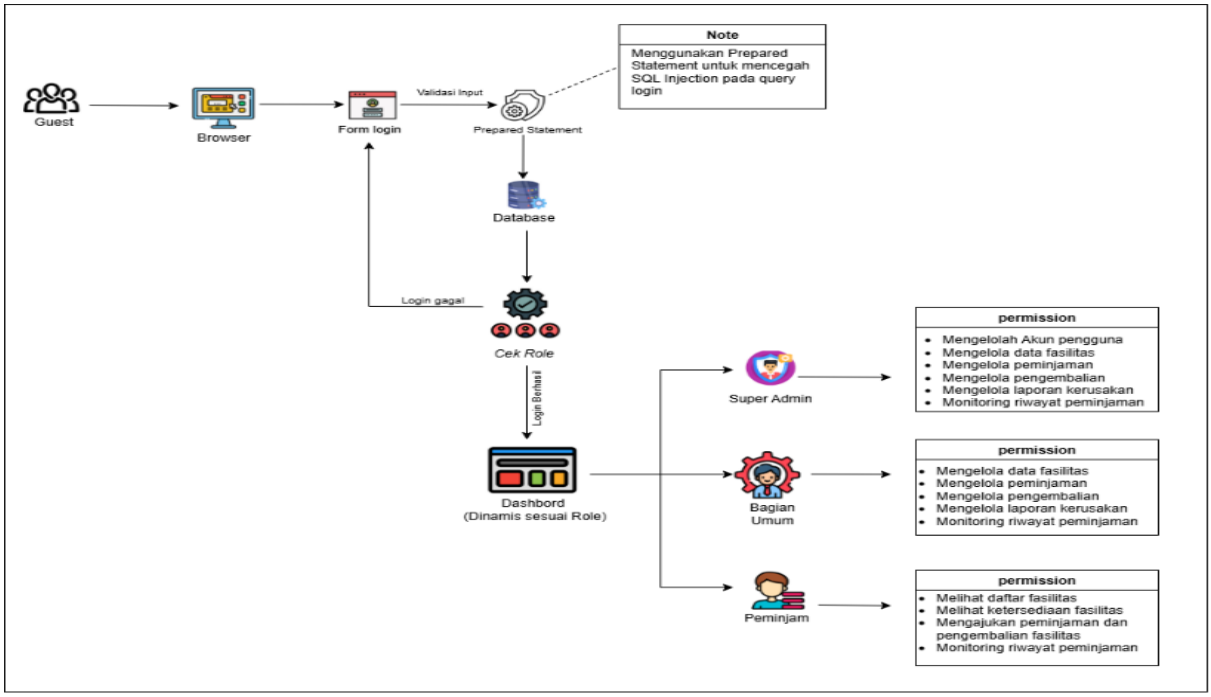


Gambar 5. Proses pelaporan kerusakan

Setelah itu, sistem mengirimkan pemberitahuan kepada *admin* agar laporan dapat ditindaklanjuti. admin meninjau laporan yang masuk dan melakukan pengecekan kondisi kerusakan secara langsung di lapangan. Berdasarkan hasil verifikasi tersebut, sistem memperbarui status laporan sesuai kondisi yang ditemukan dan menyampaikan informasi tindak lanjut kepada Peminjam sebagai akhir dari proses pelaporan kerusakan.

3.8. Implementasi

Tahap implementasi dilakukan dengan membuat aplikasi web menggunakan PHP dan MySQL. Tahap ini melibatkan penerapan langsung konsep keamanan aplikasi web dan merupakan realisasi rancangan sistem. Ini juga melibatkan penerapan seluruh interaksi dengan basis data yang melibatkan input pengguna melalui *Prepared Statement*. Untuk memastikan tidak adanya titik lemah pada sistem, sangat penting untuk memastikan bahwa *Prepared Statement* Persiapan diterapkan pada seluruh fitur. Gambar 6 menunjukkan proses di mana alur kerja sistem dibagi menjadi beberapa komponen utama.



Gambar 6. Alur *system*

Gambar 6 *memperlihatkan* alur kerja sistem yang menggambarkan proses interaksi pengguna dengan aplikasi sejak tahap autentikasi hingga akses ke halaman utama berdasarkan peran yang dimiliki. Alur dimulai saat pengguna mengakses aplikasi melalui peramban dan melakukan login dengan memasukkan username dan password, yang selanjutnya diproses menggunakan *Prepared Statement* untuk meningkatkan keamanan dan mencegah serangan *SQL Injection*. Apabila proses autentikasi tidak berhasil, sistem akan menolak akses dan menampilkan pesan kesalahan, sedangkan autentikasi yang berhasil akan dilanjutkan dengan identifikasi peran pengguna. Berdasarkan peran tersebut, sistem mengarahkan pengguna ke dashboard yang sesuai dengan hak aksesnya, yaitu Super Admin dengan kewenangan penuh dalam pengelolaan sistem, Bagian Umum yang berfokus pada pengelolaan operasional fasilitas dan peminjaman, serta Peminjam yang hanya dapat mengakses fitur terkait pengajuan dan pemantauan peminjaman fasilitas miliknya sendiri.

3.9. Pengujian Sistem

Proses pengujian dilakukan untuk mengetahui seberapa efektif sistem dari segi fungsional dan keamanan. Pengujian fungsional memastikan bahwa setiap fitur beroperasi sesuai dengan kebutuhan yang telah *ditetapkan*. Pada saat yang sama, pengujian keamanan dilakukan dengan menggunakan SQLMap untuk mensimulasikan serangan *SQL Injection* pada titik input aplikasi. Hasil pengujian digunakan untuk mengevaluasi apakah mekanisme *Prepared Statement*

yang digunakan dapat secara efektif mencegah penggunaan celah keamanan.

3.10. Pemeliharaan

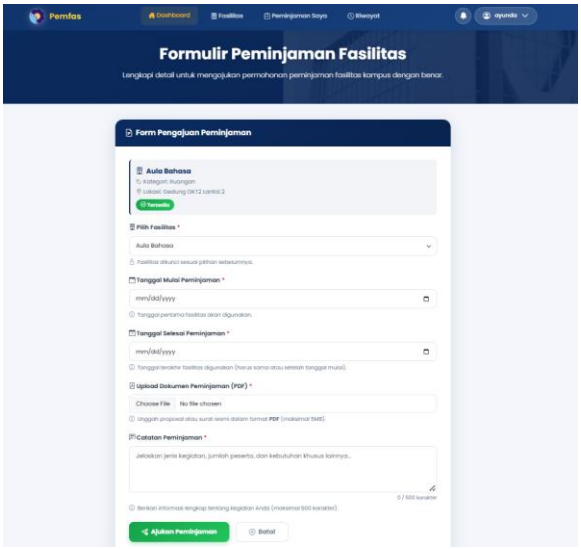
Untuk memastikan bahwa sistem tetap aman dan relevan setelah diimplementasikan, tahap pemeliharaan dilakukan untuk memperbaiki hasil pengujian, meningkatkan kinerja, dan menyesuaikannya dengan kebutuhan pengguna. Dengan demikian, sistem tidak hanya berfungsi dengan baik pada awalnya, tetapi juga tetap relevan dan aman dalam jangka panjang.

4. HASIL DAN PEMBAHASAN

Bagian ini membahas hasil perancangan dan implementasi *prepared statement* pada form input untuk mencegah *sql injection* pada aplikasi web peminjaman fasilitas. Pembahasan difokuskan pada desain sistem yang dibangun, implementasi fitur-fitur utama aplikasi, serta evaluasi kinerja sistem berdasarkan hasil pengujian fungsional yang telah dilakukan.

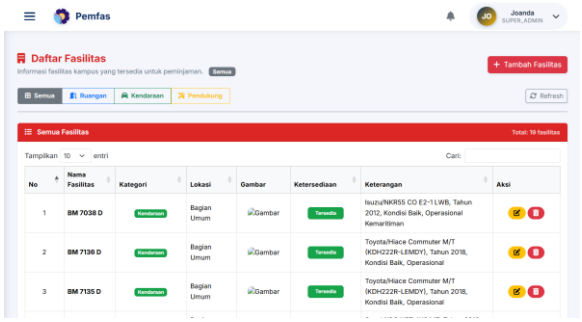
4.1. Implementasi Sistem

Hasil yang dihasilkan dari implementasi antarmuka sistem yang dirancang ditunjukkan di bagian ini.



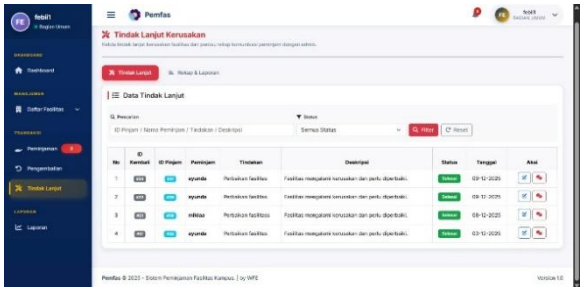
Gambar 7. Halaman Form peminjaman

Gambar 7 menunjukan bahwa peminjam dapat menggunakan formulir peminjaman fasilitas di situs web. Mereka mengisi data fasilitas, periode peminjaman, dokumen pendukung, dan catatan lainnya. Sistem akan memverifikasi data yang dimasukkan sebelum disimpan ke dalam basis data menggunakan *prepared statement* untuk memastikan keamanan dan mencegah serangan *SQL Injection*. Ini akan membuat proses pengajuan lebih aman dan efektif.



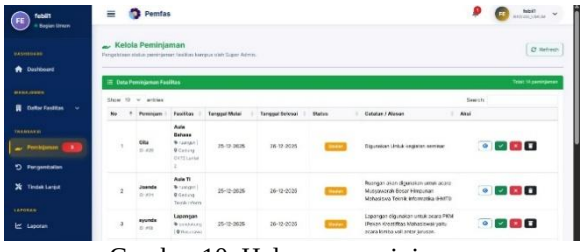
Gambar 8. Halaman daftar fasilitas peminjaman

Gambar 8 adalah halaman daftar fasilitas menampilkan data fasilitas kampus yang tersedia untuk peminjaman dalam bentuk tabel terstruktur. Sistem menyediakan fitur pencarian, penyaringan kategori, serta aksi tambah, ubah, dan hapus data sehingga pengelolaan fasilitas oleh admin dapat dilakukan secara efisien dan terpusat.



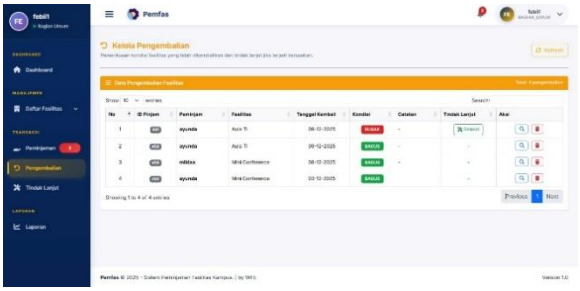
Gambar 9. Halaman tindak lanjut

Gambar 9 halaman tindak lanjut kerusakan digunakan untuk menampilkan dan mengelola laporan kerusakan fasilitas yang masuk. Informasi laporan disajikan dalam bentuk tabel yang mencakup peminjam, fasilitas, deskripsi kerusakan, status, dan tanggal. Fitur pencarian, penyaringan, serta pembaruan status memudahkan admin dalam melakukan pemantauan dan penanganan kerusakan secara terstruktur.



Gambar 10. Halaman peminjaman

Gambar 10 menunjukan Halaman kelola peminjaman yang digunakan untuk menampilkan dan mengelola seluruh data pengajuan peminjaman fasilitas yang masuk. Sistem juga menyediakan fitur pencarian dan tombol aksi untuk melihat detail, menyetujui, atau menolak pengajuan, sehingga proses pengelolaan peminjaman dapat dilakukan secara terkontrol dan efisien oleh admin.



Gambar 11. Halaman Peminjaman

Gambar 11 Halaman kelola pengembalian menampilkan data pengembalian fasilitas yang telah dipinjam dalam bentuk tabel terstruktur. Informasi yang ditampilkan termasuk peminjam, fasilitas, tanggal pengembalian, kondisi fasilitas, dan catatan tindak lanjut apabila terjadi kerusakan. Halaman ini membantu manajer memantau status pengembalian dan mengelola data untuk memastikan bahwa fasilitas dikembalikan sesuai jadwal

Tabel 1 Pengujian sistem peminjaman fasilitas

No	Fitur yang diuji	Skenario Pengujian	Hasil	Hasil Pengujian
1	Login	Pengguna mencoba SQL Injection pada field login	Sistem menolak input dan tidak mengeksekusi query berbahaya	Percobaan login tidak berhasil
2	Form Pengajuan Peminjaman	Penyisipan SQL pada form peminjaman	Input diproses sebagai data	karena database menerima input query sebagai data
3	Pengujian SQLMap	Pengujian endpoint sistem	Parameter (GET, POST, Search) tidak ditemukan celah *sql injection*	parameter – parameter yg disebutkan tidak ditemukan adanya celah

Tabel pengujian menunjukkan bahwa sistem memiliki mekanisme keamanan yang baik terhadap serangan SQL Injection. Pada fitur login, percobaan penyisipan SQL tidak berhasil karena sistem menolak input berbahaya. Pada form pengajuan peminjaman, input SQL diproses sebagai data biasa sehingga tidak memengaruhi query database. Selain itu, pengujian menggunakan SQLMap pada parameter *GET, POST, dan Search* tidak menemukan celah *SQL Injection*. Hasil ini membuktikan bahwa penerapan *Prepared Statement* efektif dalam melindungi sistem dari serangan *SQL Injection*.

5. KESIMPULAN DAN SARAN

Berdasarkan hasil perancangan, implementasi, dan pengujian yang telah dilakukan menunjukkan bahwa aplikasi web peminjaman fasilitas kampus dapat dilindungi secara signifikan dari serangan *SQL Injection* dengan menerapkan *prepared statement* pada formulir input. Pengembangan sistem menggunakan metode *Waterfall* menghasilkan aplikasi yang terstruktur dan mudah digunakan yang dapat mengintegrasikan proses peminjaman, pengembalian, dan pelaporan. Hasil pengujian keamanan menggunakan SQLMap menunjukkan bahwa integritas dan kerahasiaan data tetap terjaga karena sistem tidak memiliki celah SQL Injection. Dibandingkan dengan metode manual dan sebagian digital sebelumnya, sistem yang dikembangkan juga terbukti lebih efektif dalam mengelola fasilitas kampus. Untuk meningkatkan aksesibilitas pengguna, sistem harus dilengkapi dengan mekanisme keamanan tambahan untuk pengembangan berikutnya. Contoh mekanisme keamanan ini termasuk enkripsi data sensitif dan autentikasi multi-faktor.

DAFTAR PUSTAKA

[1] N. Nurhayati, A. Atthariq, and A. Aswandi, "Implementasi Keamanan Jaringan Dengan Metode Web Application Firewall (WAF)," *Jurnal Teknologi Rekayasa Informasi dan Komputer*, vol. 8, no. 2, p. 48, 2025, [Online]. Available: https://e-jurnal.pnl.ac.id/TRIK/article/view/7449

[2] Nova Christina Sari *et al.*, "Deteksi Kerentanan SQL Injection pada Website Menggunakan Vulnerability Assessment," *Journal Of Data Insights*, vol. 2, no. 1, pp. 9–17, 2024, doi: 10.26714/jodi.v2i1.393.

[3] OWASP Foundation, "A03:2021 - Injection," 2021.

[4] Oktarina Paramitha Sandy, "Serangan SQL Injection Jadi Aduan Siber Tertinggi Selama 2021," 2022. [Online]. Available: https://cyberthreat.id/read/13925/Serangan-SQL-Injection-Jadi-Aduan-Siber-Tertinggi-Selama-2021

[5] GoodStats, "Serangan Digital di Indonesia Tembus 139 Kasus pada Awal 2025," 2025. [Online]. Available: https://data.goodstats.id

[6] N. Sariana, F. Faisal, and A. A. Putri, "Rancang Bangun Sistem Pengelolaan Data Peminjaman dan Pengembalian Barang Berbasis Android dengan Metode Waterfall," in *Prosiding Seminar Nasional …*, 2023, pp. 249–258. [Online]. Available: https://prosiding.seminars.id/prosainteks/article/view/70%0Ahttps://prosiding.seminars.id/prosainteks/article/download/70/86

[7] N. Augusta, A. Id Hadiana, and F. R. Umbara, "Sistem Keamanan Website Dengan Multi Metode Untuk Mencegah SQL Injection," p. 315320, 2024.

[8] F. Q. Kareem *et al.*, "SQL Injection Attacks Prevention System Technology: Review," *Asian Journal of Research in Computer Science*, pp. 13–32, 2021, doi: 10.9734/ajrcos/2021/v10i330242.

[9] M. NUGRAHA and J. YASKURNIAAM, "Sistem Informasi Peminjaman Barang Berbasis

Web dengan Metode Waterfall," *MIND Journal*, vol. 5, no. 1, pp. 14–23, 2021, doi: 10.26760/mindjournal.v5i1.14-23.

[10] M. Nasereddin, A. ALKhamaiseh, M. Qasaimeh, and R. Al-Qassas, "A systematic review of detection and prevention techniques of SQL injection attacks," *Information Security Journal*, vol. 32, no. 4, pp. 252–265, 2023, doi: 10.1080/19393555.2021.1995537.

[11] R. Syarifuddin, "Development of a Web-Based Accreditation System for Study Program Management using the Waterfall Model," vol. 1, no. 2, pp. 59–66, 2025, doi: 10.47776/nuai.v1i2.1663.

[12] O. Egerton Taylor, E. Promise Sochima, V. Emmah, O. Egerton, P. Sochima, and V. Thomas, "Preventing Structured Query Language (SQL) Injection Attacks using PHP Data Object and Prepared Statement," *International Journal of Computer Science and Mathematical Theory E*, vol. 5, no. 2, pp. 1–9, 2019, [Online]. Available: www.iiardpub.org

[13] J. H. B. Johny, W. A. F. B. Nordin, N. M. B. Lahapi, and Y. B. Leau, "SQL Injection Prevention in Web Application: A Review," *Communications in Computer and Information Science*, vol. 1487 CCIS, pp. 568–585, 2021, doi: 10.1007/978-981-16-8059-5_35