



Verifikasi Fungsional Arsitektur Reservasi Tiket Berbasis *PHP-MySQL* dengan Kontrol Kuota dan Kedaluwarsa Pesanan pada Layanan *Hosting Daring Gratis*

Irfan Muzaki^{1*}, Noval Ferdian²

^{1,2} Program Studi Teknologi Rekayasa Komputer, Politeknik Pertanian Negeri Payakumbuh, Indonesia

* Penulis Korespondensi: irfanmuzaki777@gmail.com

ABSTRAK

Reservasi tiket secara manual berisiko menimbulkan ketidakkonsistenan antara jumlah pesanan, status pembayaran, dan ketersediaan kuota. Literatur telah membahas sistem pemesanan berbasis web dan layanan daring, tetapi bukti implementasi terpadu antara kontrol kuota, kedaluwarsa pesanan, dan *deployment* pada *hosting* gratis yang dilaporkan secara eksplisit masih terbatas. Penelitian ini mengevaluasi kelayakan fungsional arsitektur tersebut pada *The 4 Stairs Music Hall*. Model *waterfall* digunakan untuk pengembangan melalui tahap analisis kebutuhan, perancangan, implementasi *PHP-MySQL*, dan *deployment* pada layanan *hosting* daring gratis (*InfinityFree*). Evaluasi dilakukan dengan pengujian *black box* berurutan (satu pengguna) terhadap delapan skenario fungsional. Pada tataran rancangan, sistem menggunakan *database transaction* dan *prepared statement*, pembatalan otomatis pesanan *Pending* setelah 15 menit, *e-ticket* ber-QR code, serta *dashboard* admin; sifat *atomicity* dan ketahanan terhadap akses paralel merupakan properti rancangan yang belum diuji secara empiris pada penelitian ini. Seluruh delapan skenario menghasilkan keluaran sesuai harapan, termasuk penolakan pesanan melebihi kuota, pengembalian kuota setelah batas pembayaran, dan ekspor laporan. Aplikasi juga dapat diakses daring pada saat pengujian setelah penyesuaian kredensial basis data. Temuan ini mendukung kelayakan fungsional arsitektur dalam ruang lingkup berurutan yang diuji, tetapi belum membuktikan ketahanan konkurensi, keamanan, kinerja, maupun usability.

Kata Kunci: Reservasi Tiket, *Hosting Daring Gratis*, Pengujian *Black Box*, Kontrol Kuota, Konsistensi Data.

Riwayat Artikel

<i>Diterima:</i>	<i>Direvisi:</i>	<i>Disetujui:</i>	<i>Dipublikasikan:</i>
23 Juni 2026	30 Juni 2026	1 Juli 2026	21 Juli 2026

ABSTRACT

Manual ticket reservation may create inconsistencies among order quantities, payment status, and seat availability. Previous work has addressed web-based ordering and online services, but explicitly reported evidence of an integrated implementation combining quota control, order expiration, and deployment on free hosting remains limited. This study evaluates the functional feasibility of such an architecture for *The 4 Stairs Music Hall*. The *waterfall* model guided requirements analysis, design, *PHP-MySQL* implementation, and deployment on a free online hosting service (*InfinityFree*). Evaluation used sequential, single-user *black-box* testing across eight functional scenarios. At the design level the system uses database transactions and prepared statements, automatic cancellation of *Pending* orders after 15 minutes, QR-coded *e-tickets*, and an administrative dashboard; *atomicity* and resilience under concurrent access are design properties that were not empirically tested in this study. All eight scenarios produced the expected outputs, including rejection of orders exceeding the remaining quota, quota restoration after payment expiration, and report export. The application was also accessible online during testing after database credentials were adjusted. These findings support functional feasibility within the tested sequential scope, but do not establish concurrency resilience, security, performance, or usability.

Keywords: Ticket Reservation, Free Online Hosting, Black-Box Testing, Quota Control, Data Consistency.

Article history

Received:
23 June 2026

Revised:
30 June 2026

Accepted:
1 July 2026

Published:
21 July 2026

Copyright © 2026, Irfan Muzaki, Noval Ferdian
Nusantara Technology Journal (NTJ)
ISSN xxxx-xxxx (print), ISSN xxxx-xxxx (online)

PENDAHULUAN

Bagian ini Digitalisasi reservasi tiket tidak cukup hanya memindahkan formulir pemesanan ke media daring. Sistem juga harus menjaga konsistensi antara jumlah pesanan, status pembayaran, dan ketersediaan kuota. Ketidaksinkronan ketiga data tersebut dapat menyebabkan pencatatan ganda, pemesanan melebihi kapasitas, atau kursi tertahan oleh pesanan yang tidak diselesaikan. Permasalahan ini ditemukan pada proses reservasi *The 4 Stairs Music Hall* yang masih mengandalkan kedatangan langsung dan pesan pribadi, sehingga informasi jadwal dan sisa kuota belum tersedia melalui satu sumber data yang terpusat.

Sistem pemesanan berbasis web telah digunakan untuk menata transaksi, mempercepat pencarian data, dan menghasilkan laporan secara otomatis dibandingkan proses manual (Ibrahim, 2011; Rusdiana et al., 2018). Pengembangan sistem informasi berbasis web semacam ini juga telah banyak dilaporkan pada berbagai konteks layanan (Apriani et al., 2019; Herman & Helvin, 2022). Namun, keberhasilan membangun antarmuka dan menyimpan data belum cukup untuk menunjukkan bahwa alur reservasi kritis bekerja secara konsisten. Bukti fungsional tetap diperlukan, khususnya untuk memastikan sistem menolak pesanan yang melebihi kuota, mengembalikan kuota dari pesanan kedaluwarsa, serta mempertahankan keterhubungan data antara jadwal dan pesanan.

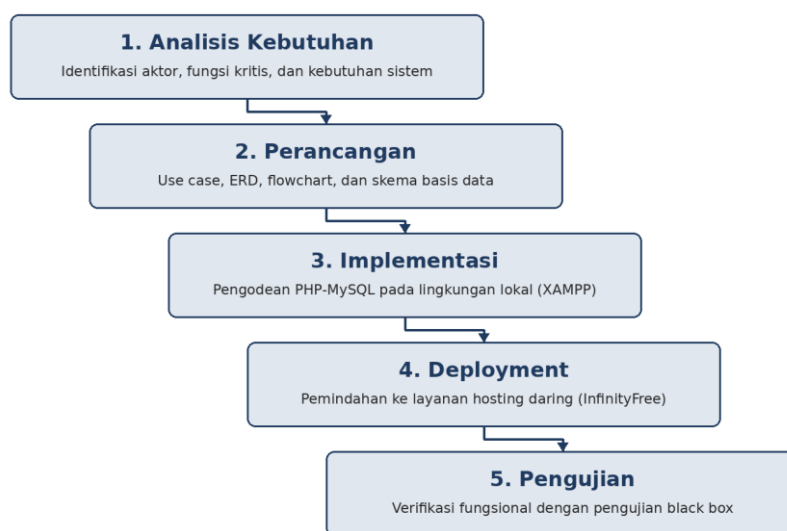
Deployment pada layanan daring memperluas akses aplikasi tanpa pengelolaan server fisik secara langsung. Konsep ini sejalan dengan karakteristik *cloud computing* yang menyediakan sumber daya komputasi melalui jaringan dengan upaya pengelolaan terbatas (Mell & Grance, 2011), serta kajian mengenai penyediaan aplikasi web pada infrastruktur daring (Alanda et al., 2022; Wati, 2023). Layanan yang digunakan pada penelitian ini, *InfinityFree*, merupakan *shared hosting* gratis sehingga tidak digunakan untuk menunjukkan elastisitas atau penskalaan sumber daya. Dalam ruang lingkup pustaka yang ditelaah, pembahasan sistem pemesanan dan *deployment* daring cenderung berdiri terpisah dan belum memberikan bukti fungsional yang dilaporkan secara eksplisit untuk integrasi kontrol kuota, kedaluwarsa pesanan, dan pengoperasian pada *hosting* gratis. Kesenjangan penelitian ini bersifat implementatif dan evaluatif, bukan kekosongan algoritmik yang bersifat universal.

Berdasarkan kesenjangan tersebut, penelitian ini menjawab dua pertanyaan: (1) apakah arsitektur reservasi PHP-MySQL yang menggabungkan kontrol kuota dan pembatalan pesanan kedaluwarsa dapat diimplementasikan dan di-*deploy* pada layanan *hosting* daring gratis; dan (2) apakah fungsi utama pengguna dan admin menghasilkan keluaran yang benar pada pengujian *black box* berurutan. Tujuan penelitian adalah merancang dan mengimplementasikan sistem, men-*deploy* aplikasi beserta basis datanya, serta memverifikasi kebenaran fungsional delapan alur kritis melalui pengujian *black box*. Klaim ilmiah yang diuji adalah bahwa arsitektur tersebut layak secara fungsional dalam ruang lingkup skenario berurutan yang ditetapkan. Kebaruan penelitian bersifat integratif dan kontekstual, yaitu dokumentasi eksplisit penerapan kontrol kuota, kedaluwarsa pesanan, dan *deployment* pada lingkungan *hosting* daring gratis; penelitian ini tidak mengklaim kebaruan algoritmik, ketahanan konkurensi, maupun keunggulan performa.

Kebaruan penelitian ini terletak pada implementasi mekanisme kontrol kuota, pembatalan otomatis pesanan berdasarkan batas waktu pembayaran, serta *deployment* aplikasi reservasi berbasis PHP-MySQL pada layanan *hosting* daring gratis yang dapat diakses publik.

METODE

Penelitian ini menggunakan pendekatan pengembangan sistem dan evaluasi fungsional. Model *waterfall* digunakan sebagai metode pengembangan, bukan sebagai metode pembuktian ilmiah, dengan lima tahap yang dikerjakan secara berurutan: analisis kebutuhan, perancangan, implementasi, *deployment*, dan pengujian. Model *waterfall* dipilih karena kebutuhan sistem relatif stabil dan terdokumentasi sejak awal sehingga sesuai dengan alur tahapan yang linier (Burhani et al., 2025; Wahid, 2020), dan model ini banyak digunakan dalam pengembangan sistem informasi berbasis web (Mallisza et al., 2022; Ningsih & Nurfauziah, 2023; Nova et al., 2022). Tahapan model *waterfall* yang digunakan ditunjukkan pada Gambar 1.



Gambar 1. Tahapan Pengembangan Sistem dengan Model *Waterfall*

Pada tahap analisis kebutuhan, diidentifikasi dua aktor, yaitu pengunjung dan admin, serta fungsi kritis berupa autentikasi, penyajian jadwal, pemesanan, validasi kuota, kedaluwarsa pembayaran, pengelolaan data, dan ekspor laporan. Tahap perancangan memvisualisasikan kebutuhan tersebut melalui *use case*, *Entity Relationship Diagram* (ERD), dan *flowchart* pemesanan. Lima tabel inti mendukung alur reservasi, yaitu jadwal, pesanan, *users*, berita, dan admin; basis data daring juga memuat tabel pendukung komposisi untuk konten musik yang tidak dianalisis sebagai bagian dari alur reservasi.

Tahap implementasi sisi server menggunakan PHP dengan ekstensi *mysqli* dan MySQL, sedangkan antarmuka menggunakan HTML, *Tailwind CSS*, dan *JavaScript*. Lingkungan lokal menggunakan XAMPP dan *phpMyAdmin* sebelum aplikasi dipindahkan ke *hosting* daring. Tahap *deployment* dilakukan pada *InfinityFree*, yaitu layanan *hosting* daring gratis berbasis *shared hosting* dengan dukungan PHP, MySQL, dan *control panel*; penelitian ini tidak menggunakannya untuk menunjukkan elastisitas atau penskalaan sumber daya sebagaimana karakteristik komputasi awan pada umumnya. Tahapannya meliputi pembuatan subdomain dan basis data, pengunggahan berkas ke direktori *htdocs*, penyesuaian kredensial koneksi, dan pemeriksaan akses aplikasi.

Tahap pengujian menggunakan pengujian *black box* berurutan (satu pengguna, satu kali jalan) terhadap delapan skenario. Pengujian *black box* menilai kesesuaian fungsi terhadap spesifikasi tanpa meninjau struktur kode internal (Jaya, 2018; Pratama et al., 2023) dan kasus ujinya dapat dirancang melalui teknik seperti *equivalence partitioning* dan *boundary value analysis* (Aryandana et al., 2020; Dobslaw et al., 2023). Kedelapan skenario dipilih untuk mencakup jalur kritis sistem, yaitu autentikasi, operasi data dasar (tambah, tampil, ubah, dan

hapus), serta dua alur yang menentukan konsistensi kuota, yakni penolakan pemesanan melebihi kuota dan pembatalan otomatis pesanan kedaluwarsa. Suatu skenario dinyatakan sesuai apabila keluaran yang diamati sama dengan keluaran yang diharapkan. Rancangan kasus uji selengkapnya disajikan pada Tabel 1. Pengujian ini merupakan verifikasi fungsional pada kondisi berurutan, bukan pengukuran reliabilitas statistik, performa, ketahanan konkurensi, keamanan, atau *usability*; pengujian akses paralel tidak dilakukan sehingga sifat *atomicity* transaksi tidak diverifikasi secara empiris.

Tabel 1. Rancangan Kasus Uji *Black Box*

Skenario	Kondisi Uji	Keluaran Diharapkan
Registrasi dan login pengguna	Akun baru didaftarkan lalu digunakan untuk masuk	Akun tersimpan, pengguna berhasil masuk
Pemesanan tiket (tambah data)	Pemesanan pada jadwal dengan sisa kuota mencukupi	Pesanan tersimpan, kuota terjual bertambah, e-ticket tampil
Menampilkan data	Memuat jadwal, berita, dan riwayat dari basis data	Data muncul sesuai isi basis data
Mengubah data (admin)	Mengubah jadwal/berita dan status bayar	Data berubah sesuai input
Menghapus data (admin)	Menghapus berita/jadwal	Data terhapus dari basis data
Pemesanan melebihi kuota	Jumlah tiket melebihi sisa kuota jadwal	Sistem menolak dan menampilkan pesan kesalahan
Kedaluwarsa pesanan	Pesanan Pending dibiarkan melewati batas 15 menit	Pesanan dibatalkan otomatis, kuota dikembalikan
Ekspor laporan (admin)	Mengekspor laporan dalam format CSV/PDF	Berkas laporan berhasil diunduh

HASIL DAN PEMBAHASAN

Hasil disusun berdasarkan bukti yang diperlukan untuk menjawab pertanyaan penelitian, yaitu rancangan mekanisme konsistensi data, hasil verifikasi fungsional, dan keberhasilan *deployment*. Tampilan antarmuka disajikan sebagai bukti implementasi, bukan sebagai ukuran utama kualitas sistem.

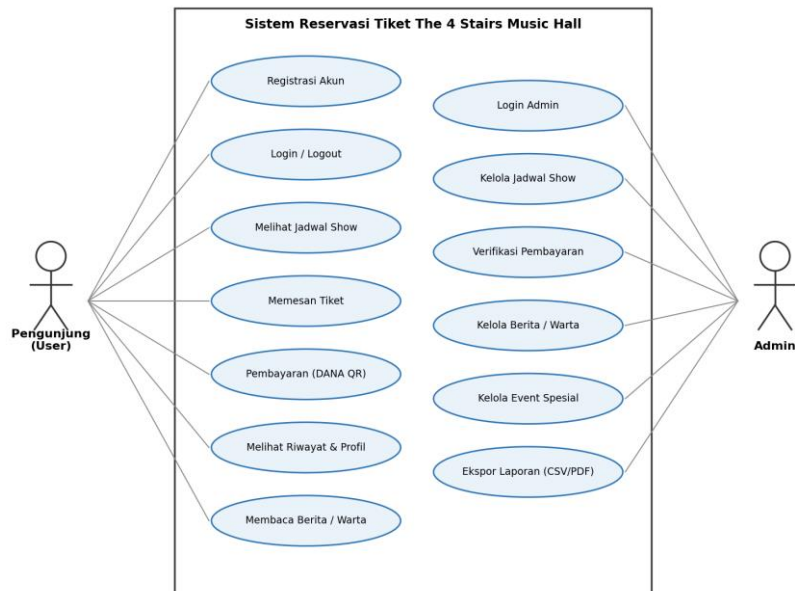
Hasil

Analisis kebutuhan menghasilkan dua aktor dengan tanggung jawab yang berbeda. Pengunjung melakukan registrasi, *login*, melihat jadwal, membuat pesanan, membaca instruksi pembayaran, dan memantau riwayat. Admin mengelola jadwal dan berita, memverifikasi pembayaran secara manual, mengelola *event* khusus, serta mengekspor laporan. Pemetaan tersebut memastikan setiap skenario pengujian memiliki aktor dan keluaran yang jelas sebagaimana ditunjukkan pada Gambar 2.

Rancangan data menggunakan lima tabel inti untuk alur reservasi. Relasi utama berbentuk satu-ke-banyak (1:N) antara jadwal dan pesanan melalui *id_jadwal* sebagai *foreign key*, sehingga setiap pesanan dapat ditelusuri ke satu jadwal tertentu. Gambar 3 membatasi ERD pada tabel inti tersebut. Pada basis data hasil *deployment* juga terdapat tabel komposisi untuk konten musik; tabel pendukung ini berada di luar ruang lingkup evaluasi reservasi dan karena itu tidak ditampilkan pada ERD inti.

Flowchart pada Gambar 4 menunjukkan titik kontrol yang menentukan konsistensi kuota. Sebelum pesanan disimpan, sistem memeriksa autentikasi, status jadwal, dan sisa kuota. Pesanan yang lolos validasi disimpan dan memperoleh batas pembayaran 15 menit. Apabila pembayaran

tidak diselesaikan, status pesanan dibatalkan dan kuota dikembalikan. Alur ini mendefinisikan perilaku yang kemudian diverifikasi melalui skenario pengujian penolakan kuota dan kedaluwarsa pesanan.

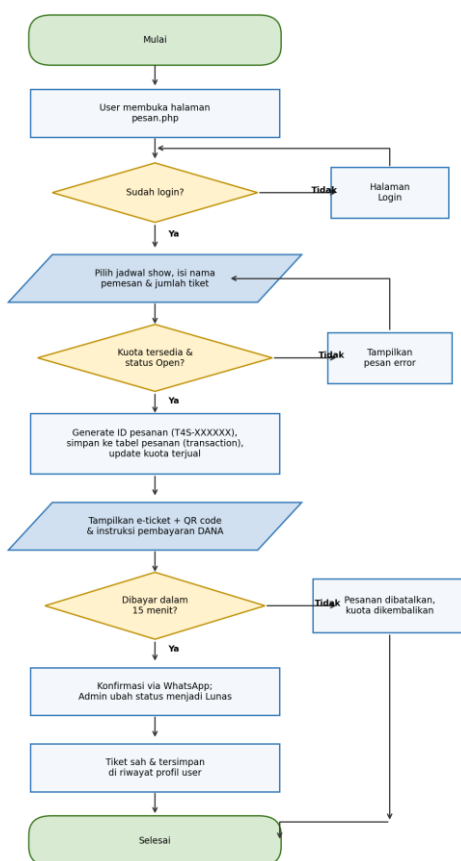


Gambar 2. Use Case Aktor dan Fungsi Utama Sistem Reservasi



Keterangan: pesanan.id_jadwal merupakan foreign key yang merujuk ke jadwal.id (ON DELETE CASCADE)

Gambar 3. Entity Relationship Diagram Lima Tabel Inti Alur Reservasi



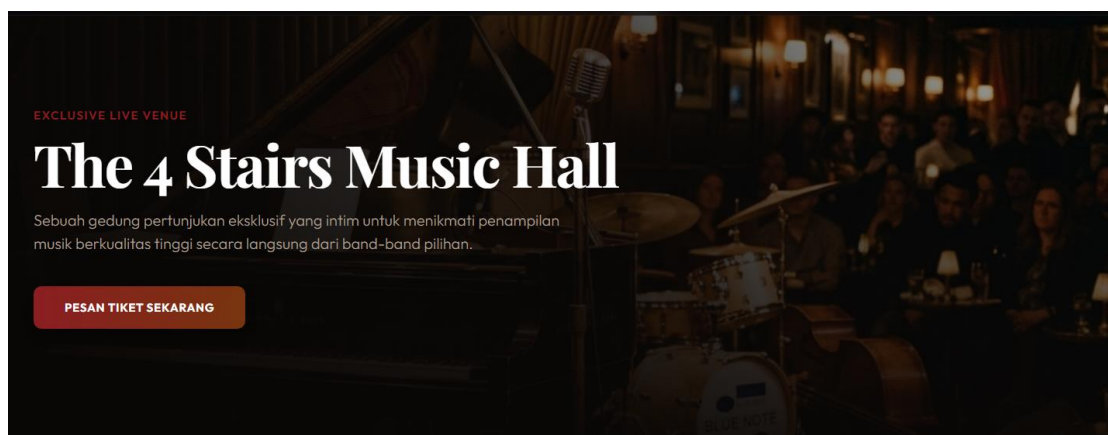
Gambar 4. Alur dan Titik Kontrol Kritis Proses Pemesanan

Basis data diimplementasikan pada *MySQL* yang berjalan di lingkungan *shared hosting InfinityFree* sebagaimana ditunjukkan pada Gambar 5. Berkas koneksi menjalankan pemeriksaan struktur tertentu, membentuk jadwal minggu berjalan apabila data belum tersedia, serta memproses pembatalan pesanan berstatus Pending yang telah melewati batas waktu 15 menit. Mekanisme ini lebih tepat dipahami sebagai inisialisasi dan pemeliharaan data berbasis *request*, bukan sebagai mekanisme self-healing dalam pengertian pemulihan kegagalan sistem secara luas.

Table	Action	Rows	Type	Collation
☐ admin		1	InnoDB	utf8mb4_general_ci
☐ berita		5	InnoDB	utf8mb4_general_ci
☐ jadwal		7	InnoDB	utf8mb4_general_ci
☐ komposisi		7	InnoDB	utf8mb4_general_ci
☐ pesanan		5	InnoDB	utf8mb4_general_ci
☐ users		2	InnoDB	utf8mb4_general_ci
6 tables	Sum	27	MyISAM	latin1_swedish_ci

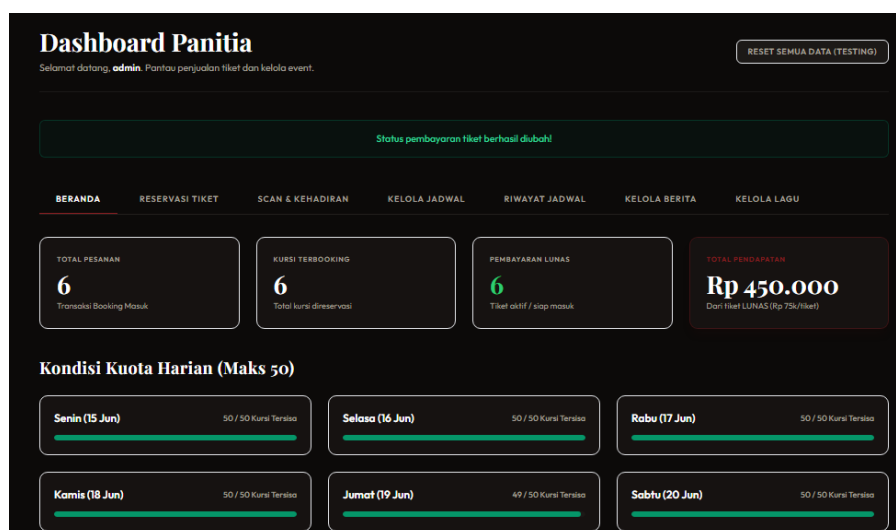
Gambar 5. Basis Data Aplikasi pada *phpMyAdmin* Lingkungan *Hosting* Daring

Pada alur pemesanan, sistem memvalidasi status jadwal dan sisa kuota sebelum menulis data. Penyimpanan pesanan dan perubahan nilai kuota ditempatkan dalam satu database transaction menggunakan operasi commit dan rollback, sedangkan prepared statement digunakan untuk menangani parameter masukan. Pada tataran rancangan, penggunaan transaksi dimaksudkan untuk menjaga atomicity proses pemesanan dan prepared statement untuk mengurangi risiko injeksi SQL. Kedua karakteristik tersebut merupakan bagian dari rancangan sistem dan belum diuji secara empiris melalui pengujian akses paralel maupun pengujian injeksi SQL pada penelitian ini. Setiap pesanan menghasilkan ID unik dan e-ticket ber-QR Code. Gambar 6 memperlihatkan implementasi aplikasi pada lingkungan *hosting* daring.



Gambar 6. Halaman Utama Aplikasi pada Lingkungan *Hosting* Daring

Dashboard admin menampilkan ringkasan pesanan, kuota terbooking, pembayaran lunas, dan pendapatan, serta menyediakan fungsi pengelolaan reservasi, jadwal, berita, dan laporan. Verifikasi pembayaran masih dilakukan secara manual oleh admin, sehingga penelitian ini tidak menguji integrasi payment gateway maupun validitas otomatis transaksi keuangan. Tampilan *dashboard* ditunjukkan pada Gambar 7.



Gambar 7. Dashboard Admin untuk Pengelolaan Reservasi dan Laporan

Pengujian *black box* secara berurutan menunjukkan bahwa seluruh delapan skenario menghasilkan keluaran yang sesuai dengan hasil yang diharapkan. Skenario mencakup autentikasi, pembuatan pesanan, penyajian data, perubahan dan penghapusan data oleh admin, penolakan pesanan yang melebihi kuota, pembatalan pesanan setelah 15 menit, dan ekspor

laporan. Dengan demikian, bukti yang tersedia mendukung terpenuhinya kebutuhan fungsional yang secara eksplisit diuji pada kondisi berurutan sebagaimana dirinci pada Tabel 2.

Tabel 2. Hasil Verifikasi Fungsional Delapan Skenario *Black Box*

Skenario dan Kondisi Uji	Keluaran Diharapkan	Keluaran Teramati	Status
Registrasi dan login pengguna; akun baru didaftarkan lalu digunakan untuk masuk	Akun tersimpan, pengguna berhasil masuk	Akun tersimpan; pengguna masuk ke sesi terautentikasi	Sesuai
Pemesanan tiket pada jadwal dengan sisa kuota mencukupi	Pesanan tersimpan, kuota terjual bertambah, e-ticket tampil	Pesanan tersimpan; nilai terjual bertambah; e-ticket ber-QR tampil	Sesuai
Menampilkan jadwal, berita, dan riwayat dari basis data	Data muncul sesuai isi basis data	Jadwal, berita, dan riwayat tampil sesuai isi basis data	Sesuai
Mengubah jadwal/berita dan status bayar oleh admin	Data berubah sesuai input	Data tersimpan berubah sesuai input admin	Sesuai
Menghapus berita/jadwal oleh admin	Data terhapus dari basis data	Baris terkait tidak lagi muncul pada basis data	Sesuai
Pemesanan dengan jumlah tiket melebihi sisa kuota	Sistem menolak dan menampilkan pesan kesalahan	Pesanan ditolak; pesan kesalahan tampil; nilai terjual tidak berubah	Sesuai
Pesanan Pending melewati batas pembayaran 15 menit	Pesanan dibatalkan otomatis, kuota dikembalikan	Status pesanan menjadi dibatalkan; sisa kuota kembali bertambah	Sesuai
Ekspor laporan (CSV/PDF) oleh admin	Berkas laporan berhasil diunduh	Berkas laporan terunduh dalam format CSV/PDF	Sesuai

Catatan: hasil pada Tabel 2 hanya merujuk pada delapan skenario berurutan yang didefinisikan dan tidak mencakup pengujian akses paralel. Kolom Keluaran Teramati merupakan hasil yang dicatat penulis pada saat pengujian; data berasal dari dokumentasi pengujian proyek, belum direplikasi secara independen, dan tidak dilengkapi data pengulangan uji, pengukuran waktu respons, maupun pengujian konkurensi. Berkas sumber dan basis data tersedia publik sehingga skenario dapat ditelusuri ulang oleh pihak lain.

Pada saat pengujian, aplikasi dapat diakses melalui <https://the4stairs.page.gd>. Jadwal dan berita dibaca dari basis data daring, sedangkan registrasi, login, pemesanan, dan dashboard menghasilkan keluaran yang sama dengan lingkungan lokal. Migrasi membutuhkan penyesuaian *host*, nama basis data, nama pengguna, dan kata sandi koneksi tanpa perubahan pada logika utama aplikasi. Hasil ini membuktikan portabilitas konfigurasi pada kasus yang diuji, tetapi tidak mengukur ketersediaan layanan jangka panjang atau skalabilitas *hosting*. Penelitian ini belum melakukan User Acceptance Test (UAT) sehingga tingkat penerimaan sistem oleh pengguna belum dievaluasi. Selain itu, belum dilakukan pengukuran usability menggunakan System Usability Scale (SUS) maupun pengujian performa seperti waktu respons atau pengujian beban, sehingga aspek tersebut menjadi ruang lingkup penelitian selanjutnya. Untuk mendukung verifikasi independen, berkas sumber aplikasi tersedia pada repositori publik (<https://github.com/Redevel00p/the4stairs>) sehingga kedelapan skenario dapat ditelusuri ulang oleh pihak lain.

Pembahasan

Hasil verifikasi pada delapan skenario menjawab pertanyaan penelitian kedua pada tingkat verifikasi fungsional berurutan: alur yang ditetapkan menghasilkan keluaran yang diharapkan. Temuan ini sejalan dengan penelitian terdahulu yang menunjukkan bahwa sistem reservasi

berbasis web dapat menata transaksi dan pelaporan (Ibrahim, 2011; Rusdiana et al., 2018) serta dengan praktik verifikasi fungsional menggunakan pengujian black box pada aplikasi berbasis web (Jaya, 2018; Pratama et al., 2023) Berbeda dengan penelitian-penelitian tersebut, penelitian ini secara khusus memverifikasi mekanisme kontrol kuota dan pembatalan otomatis pesanan dalam ruang lingkup implementasi yang diuji. Namun, penelitian ini menambahkan bukti yang lebih spesifik pada alur kontrol kuota dan kedaluwarsa pesanan. Keberhasilan kedelapan skenario tidak dapat ditafsirkan sebagai bukti keunggulan dibanding sistem lain karena tidak ada kelompok pembanding, metrik performa, atau evaluasi pengguna. Perlu pula ditekankan bahwa pada kondisi berurutan keberhasilan tersebut tidak membedakan implementasi berbasis transaksi dari implementasi non-atomik; pembuktian kontribusi transaksional karena itu memerlukan pengujian konkurensi tersendiri.

Keberhasilan skenario penolakan kuota dan pembatalan pesanan dapat dijelaskan oleh penempatan validasi sebelum penyimpanan serta pengembalian kuota ketika batas pembayaran berakhir. Pada kondisi berurutan, satu *request* diproses pada satu waktu sehingga pemeriksaan sisa kuota selalu membaca nilai terkini; karena itu kedua skenario lulus tanpa bergantung pada properti transaksi. *Database transaction* dirancang agar penambahan pesanan dan perubahan kuota diperlakukan sebagai satu kesatuan operasi, tetapi manfaat utamanya baru muncul pada akses paralel, yaitu mencegah *lost update* ketika dua *request* membaca sisa kuota yang sama secara bersamaan. Penelitian ini belum menjalankan *request* paralel atau uji beban, sehingga klaim bahwa mekanisme mencegah *race condition* pada konkurensi tinggi belum terbukti secara empiris. Pembuktiannya memerlukan eksperimen konkurensi terkendali, misalnya sejumlah *request* simultan terhadap jadwal dengan sisa kuota terbatas, dengan membandingkan kejadian *oversell* ketika transaksi diaktifkan dan dinonaktifkan.

Keberhasilan *deployment* menjawab pertanyaan penelitian pertama bahwa arsitektur PHP-MySQL dapat dioperasikan pada lingkungan *hosting* daring yang digunakan. Penyesuaian kredensial tanpa perubahan logika menunjukkan pemisahan yang cukup antara konfigurasi dan fungsi aplikasi, sejalan dengan manfaat pengelolaan sumber daya daring yang dibahas dalam literatur *cloud* (Alanda et al., 2022; Mell & Grance, 2011; Wati, 2023) Meskipun demikian, validitas eksternal penelitian terbatas pada satu aplikasi dan satu penyedia *hosting*. Kredensial yang disimpan langsung dalam berkas koneksi, verifikasi pembayaran manual, dan layanan gratis juga menjadi batas keamanan serta operasional. Penguatan bukti memerlukan uji konkurensi, pengukuran waktu respons dan ketersediaan, pengujian keamanan, evaluasi *usability* dengan pengguna, serta replikasi pada platform *hosting* lain.

Secara praktis, sistem ini memberikan manfaat langsung bagi pengelola *event* The 4 Stairs Music Hall. Admin dapat memantau status reservasi, kuota tersisa, dan pembayaran secara terpusat melalui *dashboard*, sehingga mengurangi risiko pencatatan ganda yang sebelumnya muncul akibat pengelolaan manual. Ekspor laporan dalam format CSV/PDF juga mendukung efisiensi administrasi tanpa memerlukan pengolahan data tambahan.

Bagi pengguna akhir, sistem menyediakan akses daring untuk melihat jadwal, melakukan pemesanan tiket, dan menerima *e-ticket* ber-QR Code secara otomatis, sehingga tidak perlu datang langsung untuk reservasi. Mekanisme pembatalan otomatis pesanan yang tidak dibayar dalam 15 menit juga memastikan kuota yang tidak terpakai dapat dimanfaatkan oleh calon pengunjung lain.

KESIMPULAN

Penelitian ini menunjukkan bahwa arsitektur reservasi *PHP-MySQL* dengan kontrol kuota dan kedaluwarsa pesanan dapat diimplementasikan dan di-deploy pada layanan *hosting* daring gratis yang digunakan. Pada saat pengujian, aplikasi dapat diakses secara publik setelah penyesuaian konfigurasi koneksi basis data. Dengan demikian, pertanyaan penelitian mengenai kelayakan implementasi dan *deployment* terjawab untuk konteks sistem dan penyedia *hosting* yang diuji.

Verifikasi *black box* berurutan menunjukkan kedelapan skenario menghasilkan keluaran sesuai harapan, sehingga klaim kelayakan fungsional didukung dalam batas skenario berurutan

tersebut. Kesimpulan ini tidak mencakup ketahanan terhadap akses paralel, *atomicity* transaksi, keamanan, performa, reliabilitas jangka panjang, atau kemudahan penggunaan. Sebelum sistem digunakan pada skala yang lebih besar, penelitian lanjutan perlu melakukan eksperimen konkurensi yang membandingkan kejadian *oversell* dengan dan tanpa transaksi, pengulangan uji oleh penguji independen, pengujian keamanan (termasuk percobaan injeksi dan pemindahan kredensial koneksi ke variabel lingkungan), pengukuran waktu respons dan ketersediaan, evaluasi pengguna, serta replikasi *deployment* pada infrastruktur yang berbeda, belum dilakukan User Acceptance Test (UAT), belum dilakukan System Usability Scale (SUS), belum dilakukan pengujian performa.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Bapak Agung Putra Yunanda, M.Kom. selaku dosen pengampu mata kuliah Teknologi Cloud Program Studi Teknologi Rekayasa Komputer, Politeknik Pertanian Negeri Payakumbuh, atas bimbingan dan arahan yang diberikan selama pelaksanaan proyek ini.

DAFTAR PUSTAKA

- Alanda, A., Mooduto, H. A., & Hadelina, R. (2022). Continuous integration and continuous deployment (CI/CD) for web applications on cloud infrastructures. *JITCE (Journal of Information Technology and Computer Engineering)*, 6(2), 50–56. <https://doi.org/10.25077/jitce.6.02.50-56.2022>
- Apriani, D., Aisyah, E. S., & Anggraini, L. (2019). Rancang bangun sistem informasi inventory peralatan komputer berbasis website pada PT Indonesia Toray Synthetics. *Technomedia Journal*, 4(1), 15–29. <https://doi.org/10.33050/tmj.v4i1.997>
- Aryandana, I. G. S., Permasari, A. E., & Adji, T. B. (2020). Comparing method equivalence class partitioning and boundary value analysis with study case add medicine module. *IOP Conference Series: Materials Science and Engineering*, 732(1), 012072. <https://doi.org/10.1088/1757-899X/732/1/012072>
- Burhani, I., Soderi, A., & Diantoro, K. (2025). Perbandingan metodologi SDLC waterfall dan agile dalam rencana pengembangan sistem informasi kepatuhan. *Journal of Informatics Management and Information Technology (JIMAT)*, 5(2), 147–154. <https://doi.org/10.47065/jimat.v5i2.489>
- Dobslaw, F., Feldt, R., & Gomes de Oliveira Neto, F. (2023). Automated black-box boundary value detection. *PeerJ Computer Science*, 9, e1625. <https://doi.org/10.7717/peerj-cs.1625>
- Herman, D. A., & Helvin. (2022). Perancangan dan pengembangan aplikasi jual beli properti berbasis web dengan metode SDLC (Studi kasus: PT Limitra Indonesia Mega). *Journal of Information System and Technology (JOINT)*, 3(2). <https://doi.org/10.37253/joint.v3i2.6755>
- Ibrahim, A. (2011). Sistem pemesanan tiket pesawat berbasis web. *Jurnal Sistem Informasi (JSI)*, 3(1), 320–326.
- Jaya, T. S. (2018). Pengujian aplikasi dengan metode black box testing boundary value analysis (Studi kasus: Kantor digital Politeknik Negeri Lampung). *Jurnal Informatika: Jurnal Pengembangan IT*, 3(1), 45–48. <https://doi.org/10.30591/jpit.v3i1.647>
- Mallisza, D., Hadi, H. S., & Aulia, A. T. (2022). Implementasi model waterfall dalam perancangan sistem surat perintah perjalanan dinas berbasis website dengan metode SDLC. *Jurnal Teknik, Komputer, Agroteknologi Dan Sains (Marostek)*, 1(1), 24–35. <https://doi.org/10.56248/marostek.v1i1.9>
- Mell, P., & Grance, T. (2011). The NIST definition of cloud computing. In *NIST Special Publication 800-145*. National Institute of Standards and Technology. <https://doi.org/10.6028/NIST.SP.800-145>

- Ningsih, W., & Nurfauziah, H. (2023). Perbandingan model waterfall dan metode prototype untuk pengembangan aplikasi pada sistem informasi. *Jurnal Ilmiah Metadata*, 5(1), 83–95. <https://doi.org/10.47652/metadata.v5i1.311>
- Nova, S. H., Widodo, A. P., & Warsito, B. (2022). Analisis metode agile pada pengembangan sistem informasi berbasis website: Systematic literature review. *Techno.Com*, 21(1), 139–148. <https://doi.org/10.33633/tc.v21i1.5659>
- Pratama, S. D., Lasimin, & Dadaprawira, M. N. (2023). Pengujian black box testing pada aplikasi edu digital berbasis website menggunakan metode equivalence dan boundary value. *Jurnal Teknologi Sistem Informasi Dan Sistem Komputer TGD*, 6(2), 560–569. <https://doi.org/10.53513/jsk.v6i2.8166>
- Rusdiana, L., Sugianoor, D., & Rosmiati, R. (2018). Sistem informasi pemesanan barang pada Bengkel Tawakkal Jaya berbasis web. *SINTECH (Science and Information Technology) Journal*, 1(2). <https://doi.org/10.31598/sintechjournal.v1i2.220>
- Wahid, A. A. (2020). Analisis metode waterfall untuk pengembangan sistem informasi. *Jurnal Ilmu-Ilmu Informatika Dan Manajemen STMIK*, 14(2), 1–5.
- Wati, N. (2023). Perancangan file cloud host berbasis Nextcloud. *Jurnal Teknologi Informasi Indonesia (JTII)*, 8(2), 1–6. <https://doi.org/10.30869/jtii.v8i2.1290>