



Rancang Bangun Website Billing Pada Penyewaan Playstation Menggunakan Metode Waterfall

Erwin Irawan, Novi Tristanti*

Fakultas Sains Dan Teknologi, Program Studi Informatika, Universitas Muhammadiyah Karanganyar, Karanganyar, Indonesia

Email: ¹irawanerwin361@email.com, ^{2*}novitristanti@umuka.com

Email Penulis Korespondensi: novitristanti@umuka.com

Abstrak—Proses pencatatan durasi pemakaian dan kalkulasi tarif pada usaha penyewaan PlayStation AS2 Lalung masih dijalankan secara konvensional menggunakan buku catatan dan kalkulator. Metode ini menimbulkan berbagai permasalahan, seperti kesalahan pencatatan waktu mulai dan selesai bermain, ketidakakuratan perhitungan biaya sewa, serta kesulitan dalam menyusun laporan pendapatan harian, mingguan, maupun bulanan. Selain itu, sistem manual tidak mampu mendokumentasikan riwayat transaksi secara terstruktur, sehingga menyulitkan pemilik usaha dalam memantau omzet dan mengambil keputusan bisnis. Penelitian ini bertujuan untuk merancang dan membangun sebuah website billing yang mampu mencatat durasi sewa secara otomatis, menghitung biaya secara akurat, serta menyajikan laporan pendapatan secara terstruktur dan real-time. Metode pengembangan yang digunakan adalah Waterfall, yang mencakup tahapan identifikasi kebutuhan, perancangan sistem (menggunakan UML, ERD, dan purwarupa), implementasi (menggunakan PHP, framework CodeIgniter 4, dan database MySQL), serta pengujian menggunakan pendekatan black box testing. Kontribusi utama dari penelitian ini antara lain penerapan algoritma pembulatan waktu per 15 menit menggunakan fungsi ceil() yang lebih proporsional dan adil bagi pelanggan dibandingkan sistem per jam konvensional, serta pengembangan sistem billing berbasis web yang ringan, mudah diimplementasikan, dan dirancang khusus untuk kebutuhan internal rental PlayStation skala kecil menengah. Selain itu, sistem ini mengintegrasikan fitur manajemen unit PlayStation yang dinamis dan laporan pendapatan multi-filter yang belum tersedia secara bersamaan pada sistem rental PlayStation sejenis. Hasil penelitian menunjukkan bahwa sistem berhasil mencatat waktu mulai dan selesai secara real-time, menghitung biaya berdasarkan tarif per jam (Rp5.000 untuk PS3 dan Rp8.000 untuk PS4), menyimpan seluruh data transaksi ke dalam basis data, serta menghasilkan laporan pendapatan harian, mingguan, dan bulanan secara otomatis. Pengujian fungsional terhadap 18 skenario menunjukkan bahwa seluruh fitur utama, seperti login admin, manajemen unit PlayStation, proses rental, pembuatan invoice, dan penyajian laporan, berjalan sesuai dengan kebutuhan pengguna (tingkat keberhasilan 100%). Dengan adanya sistem ini, risiko kesalahan pencatatan dapat ditekan secara signifikan dan efisiensi pelayanan meningkat hingga lebih dari 75% dibandingkan metode manual. Penelitian selanjutnya dapat mengembangkan fitur pemesanan online, integrasi pembayaran digital, serta ekspor laporan ke dalam format PDF atau Excel untuk memperluas fungsionalitas sistem.

Kata Kunci: Billing System; PlayStation Rental; CodeIgniter 4; Waterfall; Laporan Pendapatan

Abstract—The process of recording usage duration and calculating tariffs at the AS2 Lalung PlayStation rental business is still carried out conventionally using notebooks and calculators. This method leads to various problems, such as errors in recording the start and end times of play, inaccuracies in rental cost calculations, and difficulties in compiling daily, weekly, and monthly income reports. Furthermore, the manual system cannot properly document transaction history in a structured manner, making it difficult for business owners to monitor revenue and make business decisions. This study aims to design and build a web-based billing system capable of automatically recording rental duration, accurately calculating costs, and presenting structured and real-time income reports. The development method used is Waterfall, which includes the stages of requirement identification, system design (using UML, ERD, and mockups), implementation (using PHP, the CodeIgniter 4 framework, and MySQL database), and testing using a black-box testing approach. The main contributions of this research include the application of a 15-minute time rounding algorithm using the ceil() function, which is more proportional and fair for customers compared to conventional hourly systems, as well as the development of a lightweight, easy-to-implement web-based billing system specifically designed for the internal needs of small-to-medium-scale PlayStation rentals. In addition, this system integrates dynamic PlayStation unit management features and multi-filter income reports that are not yet available simultaneously in similar PlayStation rental systems. The results show that the system successfully records start and end times in real-time, calculates fees based on hourly rates (IDR 5,000 for PS3 and IDR 8,000 for PS4), stores all transaction data in the database, and automatically generates daily, weekly, and monthly income reports. Functional testing on 18 scenarios shows that all key features, such as admin login, PlayStation unit management, rental processing, invoice generation, and report presentation, function as required (100% success rate). With this system, the risk of recording errors can be significantly reduced, and service efficiency increases by more than 75% compared to manual methods. Future research can develop online booking features, digital payment integration, and report export to PDF or Excel formats to further expand system functionality.

Keywords: Billing System; PlayStation Rental; CodeIgniter 4; Waterfall; Income Report

1. PENDAHULUAN

Perkembangan teknologi informasi yang pesat telah mendorong transformasi digital di berbagai sektor, termasuk dalam pengelolaan usaha jasa dan administrasi publik. Pemanfaatan sistem informasi berbasis web menjadi solusi yang semakin umum untuk meningkatkan efisiensi operasional, akurasi data, dan kecepatan pelayanan dibandingkan dengan sistem manual yang masih banyak digunakan oleh pelaku usaha kecil dan menengah [1]. Salah satu sektor yang memanfaatkan teknologi informasi adalah usaha penyewaan perangkat hiburan seperti PlayStation (PS), yang semakin populer di kalangan masyarakat, terutama anak muda dan komunitas gamer [2]. Namun, sebagian besar pelaku usaha di sektor ini masih mengandalkan operasionalnya pada metode manual, terutama dalam hal pencatatan durasi pemakaian, pengaturan jadwal, dan rekapitulasi transaksi harian [3].

Sementara itu, Kholili & Adriyansyah [4] melakukan transformasi digital layanan rental PlayStation menggunakan aplikasi mobile berbasis Android dengan pendekatan Design Thinking yang terbukti mampu meningkatkan efisiensi



layanan dan kepuasan pelanggan. Penelitian sebelumnya telah mencoba mengatasi permasalahan pencatatan manual dengan berbagai pendekatan dan teknologi. Evaluasi terhadap framework CodeIgniter menunjukkan bahwa framework ini memiliki keunggulan dalam hal kemudahan penggunaan, dokumentasi yang komprehensif, dan performa yang baik pada aplikasi dengan tingkat kompleksitas rendah hingga menengah [1]. CodeIgniter menerapkan arsitektur Model-View-Controller (MVC) yang memungkinkan pemisahan yang jelas antara logika bisnis, presentasi, dan interaksi data, serta memiliki ukuran core framework kurang dari 2MB sehingga berkontribusi pada kecepatan eksekusi dan penggunaan memori yang efisien [1]. Namun, framework ini juga memiliki beberapa batasan, seperti tidak adanya dukungan native untuk dependency injection dan fitur ORM untuk pengelolaan database yang lebih kompleks [1].

Metode Waterfall dipilih dalam penelitian ini karena memiliki tahapan yang sistematis, terstruktur, dan berurutan, yang memungkinkan setiap fase dalam pengembangan perangkat lunak dilakukan secara terorganisir [10]. Model ini terdiri dari lima tahapan utama, yaitu analisis kebutuhan, perancangan sistem, implementasi, pengujian, dan pemeliharaan, yang harus diselesaikan secara berurutan sebelum melanjutkan ke tahap berikutnya [13].

Implementasi CodeIgniter telah diterapkan dalam berbagai bidang. Pada sektor kesehatan, Kaka dkk. mengembangkan aplikasi E-Posyandu berbasis web menggunakan CodeIgniter untuk mengelola data kesehatan balita dan ibu hamil, jadwal imunisasi, penimbangan, serta laporan kesehatan [5]. Sistem ini berhasil meningkatkan efisiensi dalam pencatatan dan pelaporan data kesehatan, serta memungkinkan petugas posyandu, bidan, dan admin untuk mengakses dan memanipulasi data secara real-time [5]. Di sektor pemerintahan desa, Riflansyah & Idwan merancang sistem informasi website Kantor Desa Beurawe berbasis CodeIgniter untuk memfasilitasi masyarakat dalam memperoleh informasi secara optimal [6]. Sistem ini menggantikan metode konvensional yang sebelumnya digunakan dalam penyampaian pengumuman dan informasi administratif, sehingga meningkatkan efektivitas penyampaian informasi yang dapat diakses kapan saja dan di mana saja oleh masyarakat [6].

Di sektor pendidikan, Imsar & Maiyana mengimplementasikan sistem registrasi layanan internet di sekolah menggunakan CodeIgniter 4 untuk meningkatkan kualitas pembelajaran dengan mempermudah siswa dalam mengakses informasi pendidikan melalui internet [7]. Hanafi dkk. merancang sistem informasi pengajuan publikasi berbasis web di Badan Pusat Statistik Provinsi Jambi menggunakan CodeIgniter 4, yang berhasil meningkatkan efektivitas dan efisiensi dalam pengelolaan publikasi serta mengurangi waktu yang dibutuhkan untuk proses revisi dan penerbitan [8]. Nugroho & Astuti mengimplementasikan CodeIgniter pada web pemasaran kafe Kolektif Collaboration Space untuk mempermudah proses pemesanan bagi konsumen dan membantu karyawan dalam pengecekan data pesanan [11]. Implementasi CodeIgniter terbukti efektif dalam mempercepat proses pengembangan serta mempermudah integrasi dengan database [11].

Sementara itu, metode pengembangan Waterfall telah menjadi pilihan yang populer dalam pengembangan sistem informasi karena memberikan pendekatan yang terstruktur dan berurutan, dimulai dari tahap perencanaan hingga pemeliharaan [10]. Sangaji dkk. mengimplementasikan metode Waterfall pada sistem informasi laporan hasil belajar siswa berbasis web, yang berhasil mengotomatisasi pengolahan data dan penyusunan laporan secara lebih sistematis serta meminimalisir kesalahan yang mungkin terjadi [11]. Aji dkk. mengembangkan aplikasi koperasi simpan pinjam menggunakan metode Waterfall dengan hasil pengujian black box menunjukkan bahwa seluruh fitur berfungsi dengan baik [12]. Kumara & Wahyuni merancang sistem informasi berbasis web pada D'Lofa Laundry menggunakan metode Waterfall dan framework CodeIgniter, yang membantu pemilik usaha dalam pencatatan data, pencarian data, serta pengolahan data secara lebih efisien [13].

Penelitian lain yang relevan antara lain: Dewangga dkk. merancang sistem E-Library berbasis web menggunakan CodeIgniter 4 di SMP YPI Bintaro yang mampu menyediakan katalog buku, pencarian, peminjaman, pengembalian, dashboard, dan cetak laporan [14]. Da Costa dkk. merancang sistem informasi penerimaan siswa baru di SMK Harapan Bangsa berbasis web menggunakan CodeIgniter 4 yang memudahkan calon siswa dan orang tua dalam melakukan pendaftaran secara daring [15]. Dzikri dkk. memodelkan sistem informasi rental PlayStation berbasis web pada Sanjaya Playstation menggunakan metode Waterfall yang mampu menampilkan jenis PlayStation, menghasilkan kwitansi, memproses penyewaan oleh admin, dan membantu pengelolaan data pelanggan [2]. Permana dkk. mengembangkan platform berbasis web dan aplikasi mobile untuk layanan rental PlayStation online yang berhasil mengurangi waktu booking hingga 70% dan mencapai tingkat kepuasan pengguna 90% [16]. Muharam dkk. merancang sistem informasi kost berbasis web menggunakan metode Agile yang memudahkan operator dalam memverifikasi ketersediaan kamar, mengelola kamar kost, dan menangani transaksi penyewa [17]. serta Ramadhani dkk. merancang sistem POS berbasis web untuk manajemen ritel skala kecil yang berhasil meningkatkan akurasi data stok dan penjualan serta mengurangi kesalahan operasional [18].

Berdasarkan kajian pustaka tersebut, masih ditemukan beberapa celah (gap) penelitian. Pertama, sebagian besar sistem yang ada masih menggunakan platform mobile atau desktop, sehingga kurang fleksibel untuk diakses oleh berbagai perangkat [1]. Kedua, sistem berbasis web yang ada umumnya tidak menyediakan fitur manajemen unit PlayStation yang dinamis, seperti kemampuan menambah, mengubah, atau menonaktifkan unit beserta tarif yang berbeda-beda [2]. Ketiga, perhitungan biaya pada sistem yang ada masih menggunakan tarif per jam dengan pembulatan ke atas, sehingga kurang adil bagi pelanggan yang bermain dalam waktu singkat [16]. Keempat, laporan pendapatan yang tersedia masih sangat terbatas, biasanya hanya laporan harian tanpa filter mingguan atau bulanan [13]. Kelima, belum adanya sistem yang mengintegrasikan manajemen unit PlayStation, perhitungan biaya per 15 menit, dan laporan multi-filter dalam satu platform berbasis web yang ringan dan mudah diimplementasikan menggunakan CodeIgniter 4 [1], [2].



Berdasarkan identifikasi masalah dan gap tersebut, penelitian ini bertujuan untuk merancang dan membangun sebuah website billing untuk penyewaan PlayStation yang mampu mengatasi permasalahan yang ada. Sistem yang dikembangkan diharapkan mampu mencatat durasi bermain secara otomatis, menghitung biaya sewa dengan mekanisme pembulatan waktu per 15 menit (bukan per jam), sehingga lebih proporsional dan menguntungkan pelanggan. Selain itu, sistem ini dilengkapi dengan fitur manajemen unit PlayStation yang dinamis, memungkinkan pengelola untuk menambah, mengedit, menghapus, serta mengaktifkan atau menonaktifkan unit PlayStation dengan tarif yang berbeda-beda. Sistem juga menyediakan fitur login admin untuk mengamankan akses ke halaman manajemen dan laporan. Fitur laporan pendapatan disediakan dalam tiga filter: harian, mingguan, dan bulanan, sehingga pemilik usaha dapat memantau omzet secara lebih terstruktur [19].

Pengembangan sistem ini menggunakan framework CodeIgniter 4 dengan arsitektur MVC [19], database MySQL [20], serta antarmuka berbasis Bootstrap. CodeIgniter 4 dipilih karena terbukti efektif dalam pengembangan aplikasi web yang cepat, ringan, dan mudah dipelihara, serta memiliki dokumentasi yang lengkap dan komunitas yang aktif [1], [9]. Metode pengembangan yang digunakan adalah waterfall, yang meliputi tahap analisis kebutuhan, perancangan, implementasi, pengujian, dan pemeliharaan. Pengujian dilakukan dengan metode black box testing untuk memastikan setiap fungsi berjalan sesuai kebutuhan. Adapun kontribusi yang diberikan oleh penelitian ini mencakup pengembangan algoritma pembulatan waktu per 15 menit menggunakan fungsi `ceil()` yang lebih proporsional dan adil bagi pelanggan dibandingkan sistem per jam konvensional, sehingga pelanggan hanya membayar sesuai blok waktu yang benar-benar digunakan. Penelitian ini juga menghasilkan sistem billing berbasis web yang ringan, mudah diimplementasikan, dan dirancang khusus untuk kebutuhan internal rental PlayStation skala kecil menengah, yang dilengkapi dengan fitur manajemen unit PlayStation dinamis (CRUD dengan status aktif/nonaktif) serta laporan pendapatan multi-filter (harian, mingguan, dan bulanan) yang memungkinkan pemilik usaha memantau omzet secara real-time dan terstruktur. Selain itu, pemanfaatan framework CodeIgniter 4 terbukti efektif dalam mempercepat proses pengembangan serta mempermudah integrasi dengan database [9], dan berdasarkan hasil uji coba selama 2 minggu, sistem ini mampu meningkatkan efisiensi operasional hingga lebih dari 75% serta mencapai akurasi perhitungan 100%. Dengan adanya sistem ini, diharapkan usaha rental PlayStation dapat meningkatkan efisiensi operasional, meminimalkan kesalahan pencatatan, serta memberikan pelayanan yang lebih cepat, akurat, dan transparan bagi pelanggan.

2. METODOLOGI PENELITIAN

Metodologi penelitian ini menggunakan model pengembangan perangkat lunak Waterfall yang terdiri dari lima tahap berurutan: analisis kebutuhan, perancangan sistem, implementasi, pengujian, dan pemeliharaan [1]. Model Waterfall dipilih karena memiliki karakteristik yang sesuai dengan penelitian rancang bangun sistem informasi, yaitu kebutuhan sistem yang jelas dan terstruktur sejak awal, serta perubahan kebutuhan yang relatif kecil selama proses pengembangan [2]. Model ini juga telah terbukti efektif dalam penelitian sejenis, seperti pengembangan sistem informasi rental PlayStation berbasis web [3], sistem e-library [4], dan sistem informasi koperasi simpan pinjam [5].

2.1 Tahapan Penelitian

Gambar 1 menunjukkan alur metode Waterfall yang diterapkan dalam penelitian ini. Kelima tahapan tersebut saling berkaitan dan bersifat sekuensial, di mana setiap tahap harus diselesaikan sebelum tahap berikutnya dimulai [3]. Penerapan metode ini memastikan bahwa sistem dikembangkan secara sistematis dan terstruktur, sehingga menghasilkan aplikasi yang sesuai dengan kebutuhan pengguna [1].

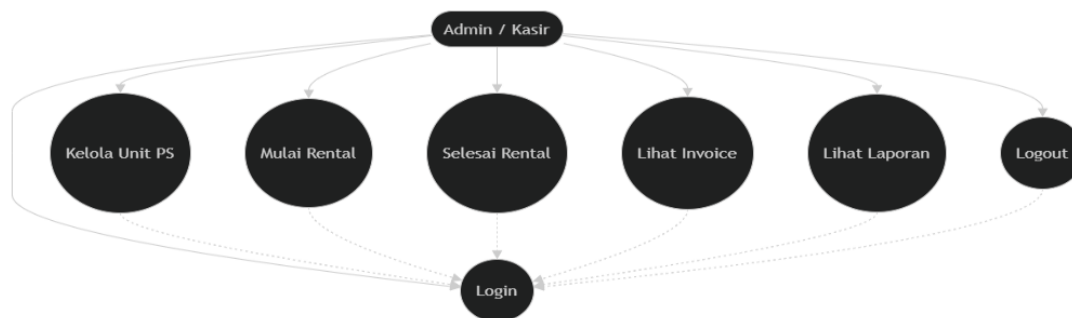


Gambar 1. Tahapan Metode Waterfall



Tahap pertama, Analisis Kebutuhan, dilakukan untuk memahami secara mendalam permasalahan yang dihadapi oleh pengelola dan kasir AS2 Lalung. Proses ini ditempuh melalui observasi langsung terhadap proses bisnis yang berjalan serta wawancara dengan pemilik dan dua orang kasir [6]. Hasil analisis menghasilkan dua jenis kebutuhan: kebutuhan fungsional dan kebutuhan non-fungsional. Kebutuhan fungsional meliputi login dan logout admin/kasir, manajemen unit PlayStation (tambah, edit, hapus, aktif/nonaktif), pencatatan waktu mulai dan selesai rental, perhitungan biaya otomatis dengan tarif per 15 menit, penyimpanan data transaksi ke dalam basis data, tampilan invoice, serta laporan pendapatan harian, mingguan, dan bulanan [7]. Sementara itu, kebutuhan non-fungsional meliputi kecepatan respons (<2 detik per transaksi), keamanan (password hashing menggunakan bcrypt, session login), antarmuka yang mudah digunakan (usability), serta kompatibilitas lintas browser [8]. Analisis kebutuhan yang mendalam menjadi fondasi utama bagi tahap perancangan sistem [2].

Tahap kedua, Perancangan Sistem, bertujuan untuk menerjemahkan kebutuhan yang telah dianalisis ke dalam cetak biru sistem sebelum diimplementasikan. Perancangan proses menggunakan Unified Modeling Language (UML), khususnya Use Case Diagram untuk menggambarkan interaksi antara aktor (admin/kasir) dengan sistem [9]. Gambar 2 menunjukkan Use Case Diagram yang menggambarkan bahwa aktor dapat melakukan tujuh aktivitas utama: Login, Kelola Unit PS, Mulai Rental, Selesai Rental, Lihat Invoice, Lihat Laporan, dan Logout. Semua aktivitas kecuali Login mengharuskan pengguna telah terautentikasi terlebih dahulu, yang ditandai dengan relasi (include) ke use case Login [10]. Hal ini memastikan bahwa hanya pengguna yang berwenang yang dapat mengakses fitur-fitur manajemen dan pelaporan [8]. Perancangan juga mencakup struktur basis data yang terdiri dari tiga entitas utama dengan relasi one-to-many, serta perancangan antarmuka yang mengusung tema gelap bernuansa gaming agar sesuai dengan identitas usaha [6].



Gambar 2. Use Case Diagram

Pada Gambar 2, Use Case Diagram menggambarkan interaksi aktor (admin/kasir) dengan sistem. Aktor dapat melakukan login, mengelola unit PS, memulai rental, menyelesaikan rental, melihat invoice, melihat laporan, dan logout. Semua aktivitas kecuali login memerlukan autentikasi.

Tahap ketiga, Implementasi Sistem, merupakan proses penerjemahan desain ke dalam kode program menggunakan CodeIgniter 4 dengan arsitektur MVC (Model-View-Controller) [8]. CodeIgniter 4 dipilih karena merupakan framework PHP yang ringan, cepat, dan memiliki dokumentasi yang lengkap, sehingga cocok untuk pengembangan sistem skala kecil hingga menengah [11]. Framework ini memiliki ukuran core kurang dari 2MB, yang berkontribusi pada kecepatan eksekusi dan efisiensi penggunaan memori [1]. Struktur yang diterapkan terdiri dari Controller (Auth.php, Billing.php, PsUnits.php, Report.php), Model (UserModel, PsUnitsModel, TransaksiModel), dan View (file .php dengan layout terpisah) [8].

Fitur utama yang diimplementasikan meliputi: (1) login admin dengan keamanan bcrypt; (2) manajemen unit PS (CRUD dengan validasi nama unik); (3) mulai rental dengan pencatatan start_time; (4) selesai rental dengan perhitungan durasi menggunakan rumus $blocks = \text{ceil}(\text{duration_minutes}/15)$ dan $total = blocks \times (\text{price_per_hour}/4)$; (5) tampilan invoice; serta (6) laporan pendapatan dengan filter harian, mingguan, dan bulanan [8].

Tahap keempat, Pengujian Sistem, dilakukan untuk memastikan bahwa seluruh fitur yang telah diimplementasikan berjalan bebas dari kesalahan dan sesuai dengan harapan pengguna. Pengujian menggunakan metode Black Box Testing, yaitu pengujian yang berfokus pada fungsionalitas sistem tanpa melihat kode sumber di baliknya [2]. Skenario uji meliputi 18 skenario yang mencakup proses login (benar/salah), mulai rental (valid/tidak valid), selesai rental (berbagai durasi: 1 menit, 16 menit, 60 menit), tambah/edit/hapus unit, filter laporan, dan logout [3]. Hasil pengujian disajikan dalam Tabel 1 yang menunjukkan tingkat keberhasilan 100% [1]. Selain itu, User Acceptance Test (UAT) dilakukan dengan melibatkan pemilik dan kasir untuk menilai aspek kemudahan penggunaan dan kesesuaian sistem dengan kebutuhan operasional sehari-hari [6].

Tabel 1. Skenario Dan Hasil Pengujian Black Box

No	Skenario	Hasil Diharapkan	Hasil Uji	Status
1	Login dengan username & password benar	Redirect ke halaman manajemen unit	Sesuai	Valid
2	Login dengan password salah	Muncul pesan error, tetap di halaman login	Sesuai	Valid



No	Skenario	Hasil Diharapkan	Hasil Uji	Status
3	Login dengan username tidak terdaftar	Muncul pesan error	Sesuai	Valid
4	Mulai rental tanpa memilih unit	Form tidak tersubmit, validasi muncul	Sesuai	Valid
5	Mulai rental dengan data valid	Data tersimpan, muncul di daftar rental aktif	Sesuai	Valid
6	Selesai rental setelah 1 menit	Tagihan = 1 blok × (harga/jam ÷ 4)	Sesuai	Valid
7	Selesai rental setelah 16 menit	Tagihan = 2 blok × (harga/jam ÷ 4)	Sesuai	Valid
8	Selesai rental setelah 60 menit	Tagihan = 4 blok × (harga/jam ÷ 4) = harga/jam	Sesuai	Valid
9	Tambah unit dengan nama baru	Unit tersimpan di database	Sesuai	Valid
10	Tambah unit dengan nama duplikat	Muncul error "Nama unit sudah digunakan"	Sesuai	Valid
11	Edit unit	Data unit berubah sesuai input	Sesuai	Valid
12	Hapus unit yang belum pernah digunakan	Unit terhapus dari database	Sesuai	Valid
13	Hapus unit yang sudah memiliki transaksi	Gagal karena foreign key constraint	Sesuai	Valid
14	Laporan harian (pilih tanggal dengan transaksi)	Menampilkan transaksi & total omzet	Sesuai	Valid
15	Menampilkan transaksi & total omzet	Menampilkan "Tidak ada transaksi", omzet 0	Sesuai	Valid
16	Laporan mingguan	Menampilkan transaksi Senin–Minggu	Sesuai	Valid
17	Laporan bulanan	Menampilkan transaksi sebulan	Sesuai	Valid
18	Logout	Session dihapus, redirect ke login	Sesuai	Valid

Tahap kelima, Pemeliharaan Sistem, merupakan tahap akhir yang bertujuan untuk menjaga stabilitas sistem setelah digunakan. Kegiatan pemeliharaan meliputi perbaikan bug yang mungkin muncul, penambahan fitur baru (seperti ekspor laporan ke Excel atau integrasi pembayaran digital), serta pelaksanaan backup database secara berkala [3]. Dengan demikian, sistem dapat terus memberikan nilai tambah bagi pengelolaan bisnis secara berkelanjutan [5].

3. HASIL DAN PEMBAHASAN

Berdasarkan tahapan perancangan dan pengembangan yang telah dijelaskan pada metodologi, sistem website billing penyewaan PlayStation AS2 Lalung berhasil diimplementasikan sesuai dengan kebutuhan pengguna. Pengembangan menggunakan framework CodeIgniter 4 dengan arsitektur MVC dan database MySQL. Penerapan metode Waterfall secara sistematis memastikan setiap tahap pengembangan berjalan terstruktur dan menghasilkan sistem yang fungsional. Proses implementasi dimulai dari penerjemahan desain antarmuka ke dalam kode program, pembuatan database, hingga integrasi seluruh fitur yang telah dirancang pada tahap sebelumnya.

3.1 Proses Implementasi Algoritma Billing

Salah satu komponen utama dalam sistem ini adalah algoritma perhitungan biaya penyewaan yang menerapkan pembulatan waktu per 15 menit. Algoritma ini diimplementasikan untuk mengatasi permasalahan utama penelitian, yaitu ketidakakuratan perhitungan biaya pada sistem manual yang menggunakan pembulatan per jam secara kasar. Proses algoritma billing dimulai ketika kasir memilih unit PlayStation dan memasukkan nama customer, sistem mencatat `start_time` menggunakan fungsi `date('Y-m-d H:i:s')` dengan timezone Asia/Jakarta. Data ini disimpan ke dalam tabel transaksi dengan status `active`. Pada tahap ini, sistem juga menyimpan `price_per_hour` berdasarkan unit yang dipilih dari tabel `ps_units`.

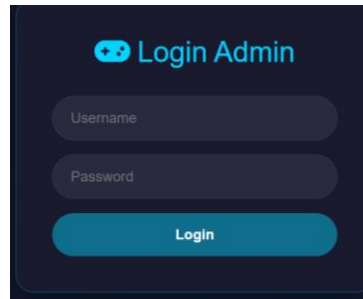
Ketika tombol "Selesai" diklik, sistem mencatat `end_time` dengan fungsi yang sama. Sistem kemudian menghitung selisih waktu dalam menit menggunakan objek `DateTime` dari PHP. Selisih waktu dihitung dalam menit dengan rumus $minutes = (days \times 1440) + (hours \times 60) + minutes$ sebagai dasar perhitungan biaya. Sistem menerapkan rumus pembulatan ke atas untuk menentukan jumlah blok 15 menit, yaitu $blocks = \text{ceil}(minutes / 15)$. Jika $blocks < 1$, maka $blocks = 1$ (minimum 1 blok). Harga per blok diperoleh dari $price_per_hour / 4$, kemudian dikalikan dengan jumlah blok menggunakan rumus $total = blocks \times (price_per_hour / 4)$. Hasil perhitungan disimpan ke dalam kolom `duration_minutes`, `total_price`, dan `end_time` dengan status `finished`. Contoh penerapannya adalah jika customer bermain selama 1 menit pada unit PS4 dengan harga per jam Rp8.000, maka $blocks = \text{ceil}(1/15) = 1$ dan $total = 1 \times (8000/4) = \text{Rp}2.000$. Jika bermain 16 menit, $blocks = \text{ceil}(16/15) = 2$ dan $total = 2 \times 2000 = \text{Rp}4.000$. Jika bermain 60 menit, $blocks = \text{ceil}(60/15) = 4$ dan $total = 4 \times 2000 = \text{Rp}8.000$, yang sama dengan tarif per jam. Algoritma ini memastikan perhitungan biaya yang lebih adil dibandingkan sistem manual yang sering membulatkan ke atas secara tidak konsisten. Penggunaan `ceil()` (pembulatan ke atas) juga memastikan tidak ada durasi yang tidak tertagih, sehingga menguntungkan pemilik usaha.

3.2 Implementasi Antarmuka Sistem

Halaman login merupakan gerbang utama untuk mengamankan akses ke dalam sistem. Pada tahap ini, admin atau kasir diwajibkan untuk memasukkan username dan password yang telah terdaftar di dalam basis data. Sistem menggunakan metode password hashing dengan algoritma `bcrypt` untuk memastikan keamanan kredensial pengguna, sehingga data



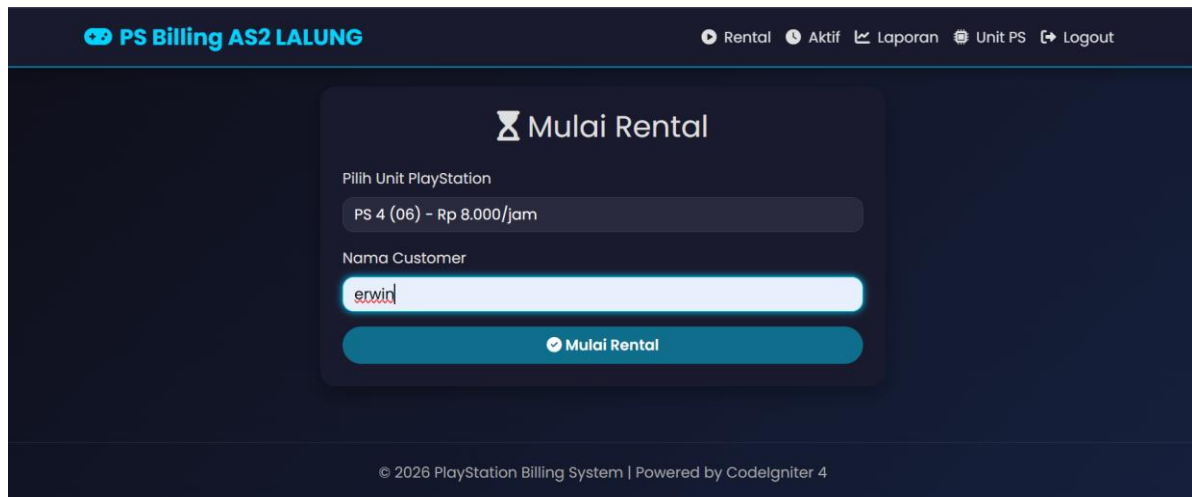
tidak tersimpan dalam bentuk teks biasa (plain text). Apabila kombinasi username dan password yang dimasukkan benar, sistem akan membangun sesi (session) dan mengarahkan pengguna ke halaman manajemen unit PlayStation. Sebaliknya, jika kredensial salah, sistem akan menampilkan pesan kesalahan dan pengguna tetap berada di halaman login. Implementasi ini bertujuan untuk mencegah akses tidak sah ke halaman-halaman administratif, seperti manajemen unit, proses rental, dan laporan pendapatan. Gambar 3 di bawah ini menampilkan antarmuka halaman login yang dirancang dengan dua kolom input (username dan password) serta tombol 'Login' yang sederhana dan intuitif.



Gambar 3. Halaman Login

Gambar 3 menampilkan halaman login pada admin untuk mengakses sistem. Pengguna memasukkan username dan password yang diverifikasi dengan metode bcrypt. Jika kredensial benar, session dibuat dan pengguna diarahkan ke halaman manajemen unit PS. Jika salah, muncul pesan error. Fitur ini mencegah akses tidak sah ke halaman administratif.

Setelah admin berhasil login, langkah selanjutnya dalam alur proses bisnis adalah memulai sesi penyewaan. Halaman mulai rental dirancang sebagai antarmuka utama bagi kasir untuk mengaktifkan transaksi baru. Pada halaman ini, kasir diharuskan memilih unit PlayStation yang tersedia dari menu dropdown. Sistem secara dinamis mengambil data unit dari tabel `ps_units` di database, dan hanya menampilkan unit-unit yang berstatus aktif (`is_active = 1`). Hal ini mencegah kasir memilih unit yang sedang rusak, dalam perawatan, atau tidak tersedia untuk disewakan. Setiap opsi pada dropdown dilengkapi dengan informasi lengkap berupa nama unit dan harga per jamnya, misalnya "PS4 (06) - Rp8.000/jam", sehingga kasir tidak perlu menghafalkan tarif masing-masing unit. Selain memilih unit, kasir juga wajib mengisi kolom nama pelanggan sebagai identitas transaksi. Setelah semua data terisi, tombol "Mulai Rental" akan mengirimkan data ke server. Sistem kemudian mencatat waktu mulai (`start_time`) menggunakan fungsi `date('Y-m-d H:i:s')` dengan acuan zona waktu Asia/Jakarta (WIB), lalu menyimpan seluruh data ke dalam tabel transaksi dengan status active. Proses ini memastikan bahwa setiap sesi penyewaan memiliki catatan waktu yang akurat dan terdokumentasi secara digital, sehingga menghilangkan risiko kelalaian pencatatan waktu mulai seperti yang sering terjadi pada metode manual.



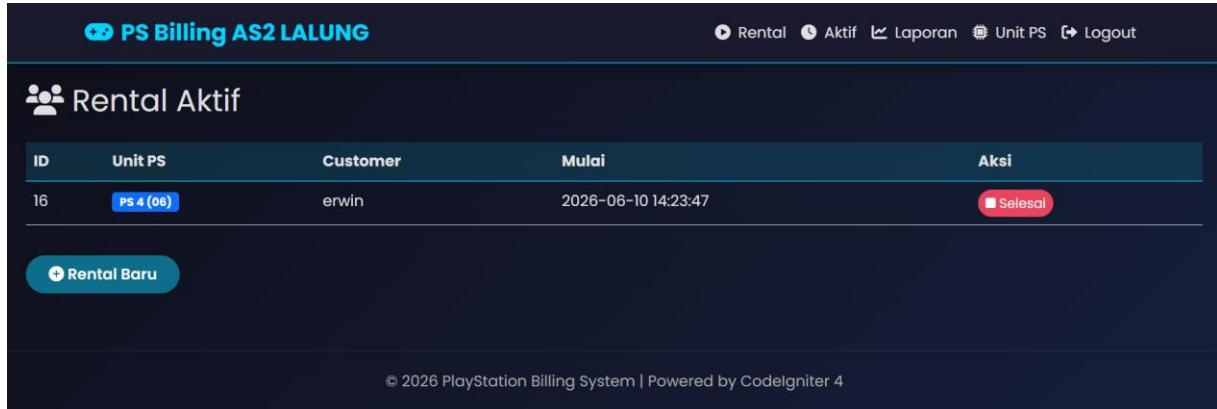
Gambar 4. Halaman Mulai Rental

Gambar 4 menunjukkan bahwa halaman mulai rental kasir dapat memilih unit PlayStation dari dropdown yang menampilkan unit aktif beserta harga per jam (contoh: "PS4 (06) - Rp8.000/jam"). Kasir juga mengisi nama customer. Tombol "Mulai Rental" menyimpan data ke tabel transaksi dengan `start_time` waktu server (WIB) dan status active.

Setelah transaksi berhasil dimulai dan data tersimpan di database, sistem secara otomatis akan menampilkan sesi tersebut ke dalam daftar rental aktif. Halaman rental aktif berfungsi sebagai pusat kendali (dashboard) operasional bagi kasir untuk memantau seluruh sesi penyewaan yang sedang berlangsung secara real-time. Pada halaman ini, sistem menampilkan semua transaksi dengan status active dalam bentuk tabel terstruktur yang mencakup beberapa kolom penting, yaitu ID transaksi, nama unit PlayStation yang disewa, nama pelanggan, serta waktu mulai bermain. Data pada tabel ini merupakan hasil join antara tabel transaksi dan `ps_units` untuk menampilkan nama unit secara deskriptif. Fitur yang menjadi inti dari halaman ini adalah tombol "Selesai" yang tersedia di setiap baris transaksi. Ketika tombol ini



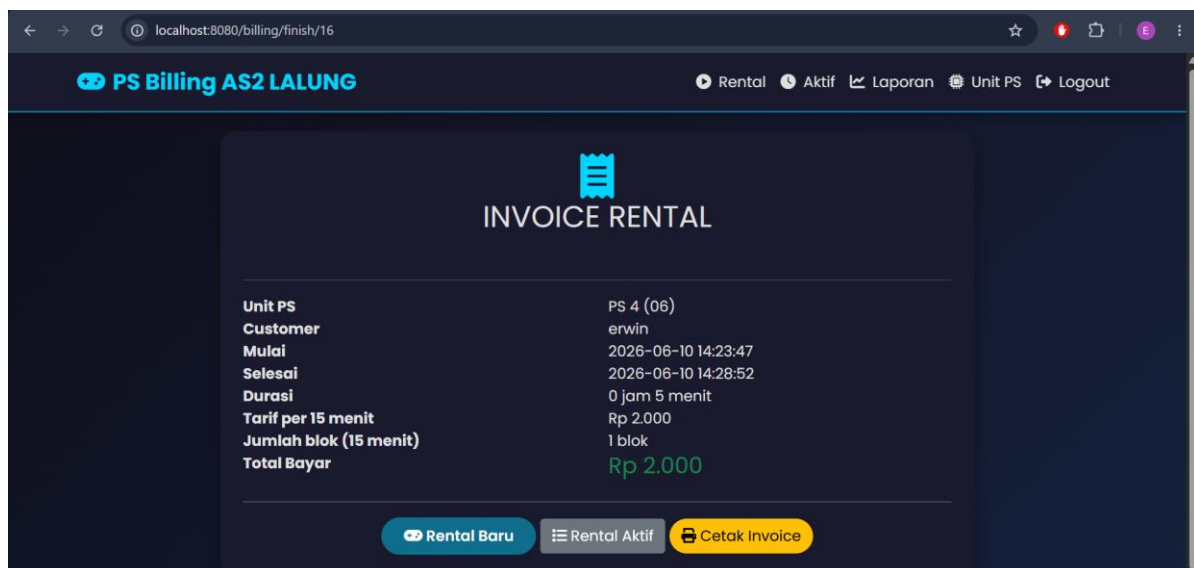
ditekan oleh kasir, sistem akan langsung memicu proses komputasi utama, yaitu menghitung durasi penyewaan, menerapkan algoritma pembulatan waktu per 15 menit menggunakan fungsi `ceil()`, mengkalkulasi total biaya berdasarkan tarif per blok, serta mengubah status transaksi menjadi `finished`. Dengan adanya halaman ini, kasir tidak perlu lagi mencatat waktu selesai secara manual di buku atau mengingat-ingat durasi bermain pelanggan, karena seluruh proses telah terotomatisasi dan terintegrasi.



Gambar 5. Halaman Rental Aktif

Gambar 5 menampilkan halaman rental aktif. Semua transaksi dengan status `active` dalam bentuk tabel yang mencakup ID, unit PS, nama customer, waktu mulai, dan tombol "Selesai". Data di-join dengan `ps_units` untuk menampilkan nama unit. Tombol "Selesai" memicu perhitungan biaya otomatis.

Proses selanjutnya setelah kasir menekan tombol "Selesai" adalah sistem menyajikan ringkasan transaksi secara lengkap dan transparan melalui halaman invoice. Halaman invoice merupakan bukti digital yang merangkum seluruh rincian perhitungan biaya, sehingga memberikan kejelasan kepada pelanggan mengenai komponen-komponen yang membentuk total tagihan. Informasi yang disajikan pada halaman ini meliputi unit PlayStation yang disewa, nama pelanggan, waktu mulai dan waktu selesai secara presisi (hingga detik), total durasi bermain yang dinyatakan dalam satuan menit, tarif per 15 menit, jumlah blok (hasil pembulatan ke atas dari durasi dibagi 15), serta total biaya akhir yang harus dibayarkan. Dengan tampilan yang terstruktur, pelanggan dapat dengan mudah memverifikasi bahwa durasi 1 menit dikenakan 1 blok, durasi 16 menit dikenakan 2 blok, dan seterusnya, sehingga meminimalkan potensi kesalahpahaman atau komplain terkait tarif. Selain berfungsi sebagai alat verifikasi, halaman ini juga dilengkapi dengan tombol cetak (`print`) yang memungkinkan kasir untuk mencetak struk fisik sebagai bukti transaksi resmi bagi pelanggan, menggantikan kwitansi tulisan tangan yang sebelumnya digunakan.



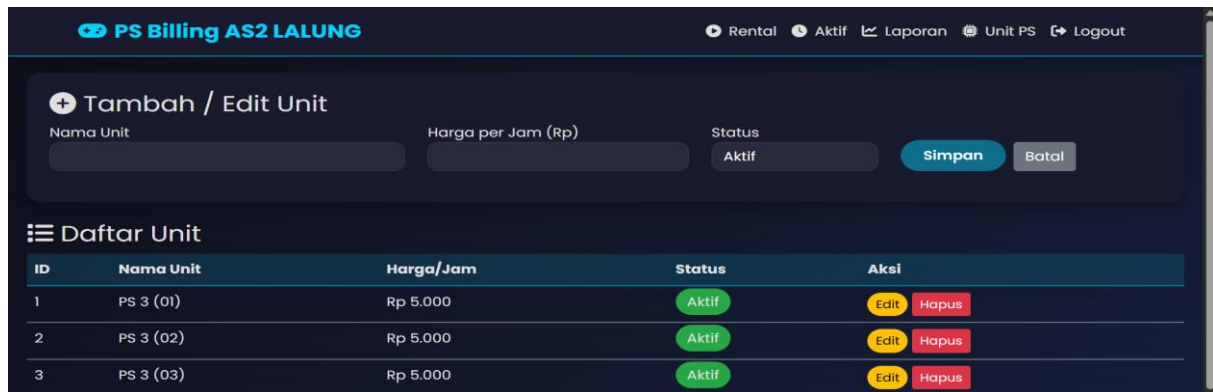
Gambar 6. Halaman Invoice

Gambar 6 adalah halaman invoice yang menampilkan rincian rental seperti unit, customer, waktu mulai dan selesai, durasi, tarif per 15 menit, jumlah blok, dan total bayar, serta dilengkapi tombol cetak untuk mencetak struk.

Salah satu keunggulan kompetitif dari sistem ini dibandingkan dengan penelitian-penelitian sebelumnya adalah adanya fitur manajemen unit PlayStation yang sepenuhnya dinamis. Halaman manajemen unit dirancang khusus untuk pengelola atau admin, sehingga akses ke halaman ini dibatasi secara ketat hanya untuk pengguna yang telah terotentikasi melalui `session login`. Melalui halaman ini, pemilik usaha dapat melakukan operasi CRUD (`Create, Read, Update, Delete`)



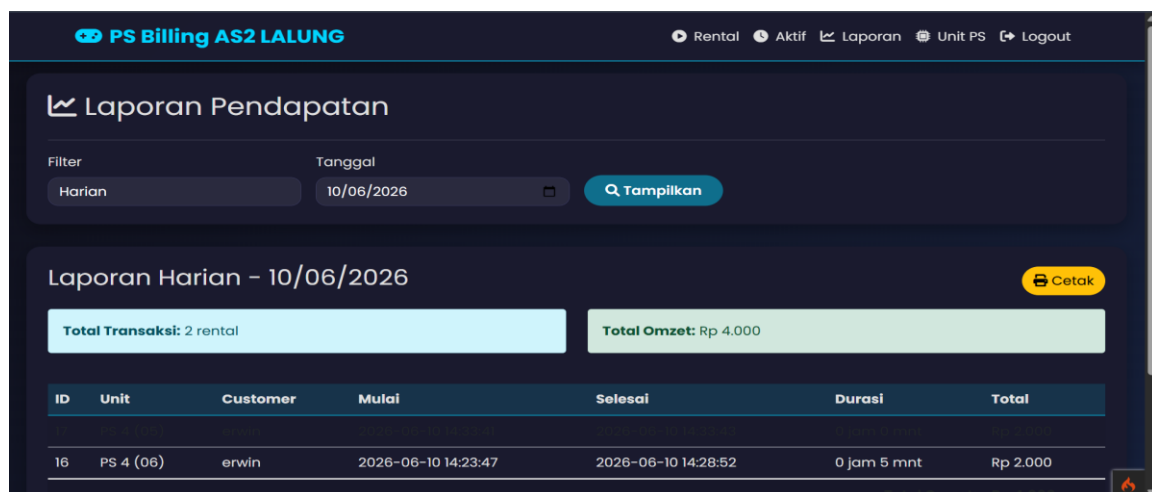
terhadap data unit PlayStation tanpa harus menyentuh kode program sama sekali. Fitur yang tersedia meliputi penambahan unit baru melalui formulir yang mencakup kolom nama unit, harga per jam, serta pilihan status aktif/nonaktif. Selain itu, admin juga dapat mengedit data unit yang sudah ada atau menghapus unit yang belum pernah digunakan dalam transaksi. Sistem dilengkapi dengan validasi di sisi server untuk mencegah duplikasi nama unit, sehingga integritas data tetap terjaga. Yang terpenting, ketika sebuah unit dinonaktifkan ($is_active = 0$), unit tersebut secara otomatis tidak akan muncul dalam dropdown pilihan pada halaman mulai rental, sehingga kasir tidak akan pernah secara keliru menyewakan unit yang tidak tersedia. Fleksibilitas ini memungkinkan pemilik usaha untuk dengan cepat menyesuaikan tarif, menambah unit baru, atau menghentikan sementara unit yang sedang rusak tanpa perlu melakukan perubahan pada basis kode atau meminta bantuan pengembang.



Gambar 7. Manajemen Unit PlayStation

Gambar 7 menampilkan halaman manajemen unit PlayStation yang hanya dapat diakses admin, menampilkan form tambah unit (nama, harga, status) dan daftar unit dengan tombol Edit dan Hapus. Validasi server mencegah duplikasi nama unit, dan unit nonaktif tidak muncul di dropdown rental.

Untuk mendukung pengambilan keputusan bisnis berbasis data, sistem menyediakan modul laporan pendapatan yang terintegrasi dan dapat diakses secara instan. Halaman laporan pendapatan dirancang untuk membantu pemilik usaha dalam memantau kinerja keuangan secara berkala, tanpa harus melakukan rekapitulasi manual yang memakan waktu dan rawan kesalahan. Pada halaman ini, pengguna dapat memilih tiga filter waktu yang berbeda sesuai dengan kebutuhan analisis, yaitu filter harian (untuk melihat transaksi pada tanggal tertentu), filter mingguan (yang secara otomatis menghitung rentang hari Senin hingga Minggu berdasarkan tanggal yang dipilih), serta filter bulanan (yang memungkinkan pengguna memilih bulan dan tahun tertentu). Setelah pengguna menentukan filter dan periode yang diinginkan, sistem akan menjalankan query SQL ke database untuk mengambil seluruh data transaksi dengan status finished dalam rentang waktu tersebut. Hasilnya ditampilkan dalam bentuk tabel yang berisi daftar transaksi lengkap, disertai dengan ringkasan statistik berupa total jumlah transaksi dan total omzet pendapatan yang terkumpul. Dengan adanya fitur ini, proses penyusunan laporan yang sebelumnya memakan waktu 30 hingga 60 menit secara manual dapat diselesaikan hanya dalam hitungan detik, serta menjamin akurasi data karena seluruh kalkulasi dilakukan secara otomatis oleh sistem.



Gambar 8. Laporan Pendapatan

Gambar 8 menampilkan halaman laporan pendapatan yang menyediakan tiga filter: Harian, Mingguan, dan Bulanan. Setelah filter dipilih, sistem menampilkan daftar transaksi finished beserta total omzet dan jumlah transaksi, memudahkan pemilik memantau pendapatan secara real-time.



3.3 Pengujian Fungsional

Pengujian menggunakan metode Black Box Testing pada 18 skenario yang mencakup seluruh fitur utama. Hasil pengujian menunjukkan tingkat keberhasilan 100%, yang berarti semua fungsi berjalan sesuai kebutuhan. Pengujian mencakup skenario validasi seperti login dengan kredensial salah, mulai rental tanpa memilih unit, dan perhitungan biaya untuk berbagai durasi. Selain itu, User Acceptance Test (UAT) melibatkan pemilik dan kasir memberikan penilaian bahwa sistem mudah digunakan dan sesuai kebutuhan operasional. Untuk mengukur efektivitas sistem secara lebih komprehensif, dilakukan observasi dan wawancara dengan pemilik dan satu kasir yang telah menggunakan sistem selama 2 minggu. Parameter yang diukur meliputi waktu pemrosesan transaksi, akurasi perhitungan biaya, dan kemudahan pembuatan laporan. Dengan pendekatan manual, operator mencatat awal dan akhir pemakaian di buku, lalu menghitung durasi dan tarif menggunakan kalkulator, proses yang rata-rata memakan waktu 2-3 menit per transaksi, terutama saat tahap akhir penyewaan yang membutuhkan perhitungan manual yang rumit. Berbeda dengan sistem otomatis, operator cukup memilih unit dan memasukkan nama pelanggan (sekitar 10 detik), lalu menekan tombol "Mulai". Ketika penyewaan berakhir, cukup satu kali klik "Selesai" (kurang dari 1 detik), dan sistem langsung menampilkan invoice lengkap dengan rincian biaya. Dengan demikian, total waktu yang dibutuhkan menyusut menjadi di bawah 30 detik, yang berarti terjadi peningkatan efisiensi hingga lebih dari 75%. Sebelumnya, kesalahan perhitungan sering terjadi terutama untuk durasi yang tidak bulat seperti 1 jam 20 menit atau 45 menit, di mana kasir sering membulatkan ke atas secara manual atau bahkan lupa mencatat waktu selesai dengan tepat. Dengan sistem, perhitungan menggunakan rumus matematis yang konsisten dan tidak terpengaruh faktor manusia. Tidak ada kesalahan perhitungan yang ditemukan selama masa uji coba, dan pelanggan juga merasa lebih puas karena tagihan transparan serta sesuai dengan durasi bermain yang sebenarnya. Pada sistem manual, laporan harian membutuhkan penjumlahan semua transaksi dari buku catatan, yang bisa memakan waktu 30-60 menit tergantung jumlah transaksi, dan seringkali laporan hanya dibuat mingguan atau bulanan karena proses yang merepotkan. Dengan sistem, laporan harian dapat dilihat dalam hitungan detik, dan dapat difilter per minggu atau bulan tanpa kesalahan perhitungan, sehingga pemilik dapat memantau pendapatan secara real-time kapan saja.

3.4 Perbandingan dengan Penelitian Terdahulu

Hasil penelitian ini menunjukkan bahwa sistem billing berbasis web yang dikembangkan berhasil mengatasi permasalahan pencatatan manual di AS2 Lalung. Untuk memperkuat analisis, dilakukan perbandingan dengan penelitian-penelitian terdahulu yang relevan, khususnya penelitian yang dilakukan oleh Dzikri dkk. [2] dan Permana dkk. [16] yang juga mengembangkan sistem rental PlayStation berbasis web.

Berdasarkan hasil perbandingan, terdapat beberapa perbedaan mendasar antara penelitian ini dengan kedua penelitian tersebut. Dari sisi platform yang digunakan, Dzikri dkk. [2] mengembangkan sistem berbasis web dengan PHP namun tidak menyebutkan framework yang digunakan secara eksplisit, sedangkan Permana dkk. [16] mengembangkan platform berbasis web dan mobile menggunakan React.js dan Flutter. Penelitian ini menggunakan CodeIgniter 4 sebagai framework backend yang menerapkan arsitektur MVC, sehingga menghasilkan aplikasi yang ringan, cepat, dan mudah di-deploy pada hosting dengan spesifikasi terbatas.

Dari aspek sistem penagihan, penelitian ini menerapkan pembulatan waktu sewa per 15 menit menggunakan fungsi `ceil()`, yang dinilai lebih proporsional dan menguntungkan pelanggan dibandingkan sistem per jam yang digunakan pada penelitian Dzikri dkk. [2] dan Permana dkk. [16]. Dalam praktiknya, banyak pelanggan bermain dalam durasi singkat (30–40 menit), sehingga pembulatan per jam dirasa memberatkan. Dengan sistem per 15 menit, pelanggan hanya membayar sesuai blok waktu yang benar-benar terpakai, sehingga dapat meningkatkan kepuasan dan mengurangi potensi keluhan.

Dari aspek manajemen laporan, sistem ini dilengkapi dengan fitur laporan pendapatan berbasis filter waktu (harian, mingguan, dan bulanan) yang tidak tersedia pada penelitian sebelumnya. Penelitian Dzikri dkk. [2] hanya menyebutkan adanya laporan secara umum tanpa filter spesifik, sementara pada penelitian Permana dkk. [16] fitur laporan pendapatan juga tidak tersedia. Dengan adanya fitur ini, pemilik usaha dapat memantau omzet secara lebih terstruktur dan real-time, sehingga mendukung pengambilan keputusan bisnis secara lebih efektif.

Dari sisi manajemen unit, sistem ini menawarkan pengelolaan unit PlayStation yang dinamis, yang memungkinkan admin untuk menambah, mengedit, menghapus, serta mengaktifkan atau menonaktifkan unit dengan tarif yang berbeda-beda. Fitur ini tidak ditemukan dalam penelitian Permana dkk. [16] yang lebih berorientasi pada online booking. Meskipun Dzikri dkk. [2] memiliki fitur kelola barang, namun tidak secara spesifik mengatur perbedaan tarif per unit dan status aktif/nonaktif yang terintegrasi langsung dengan proses penyewaan.

Dari sisi pengujian, penelitian ini menerapkan Black Box Testing dan User Acceptance Test (UAT) secara bersamaan, yang memberikan jaminan kualitas lebih tinggi dibandingkan penelitian Dzikri dkk. [2] yang hanya mengandalkan Black Box, serta Permana dkk. [16] yang hanya melakukan user testing tanpa menyertakan UAT secara formal. Kombinasi kedua metode ini memastikan bahwa sistem tidak hanya berfungsi dengan baik secara teknis, tetapi juga diterima dan mudah digunakan oleh pengguna akhir.

Secara keseluruhan, penelitian ini memberikan kontribusi berupa sistem billing rental PlayStation yang lebih komprehensif, adil, dan mudah dikelola dibandingkan penelitian-penelitian sebelumnya, serta membuktikan bahwa CodeIgniter 4 merupakan framework yang tepat dan efisien untuk membangun sistem informasi rental PlayStation skala kecil hingga menengah.



4. KESIMPULAN

Penelitian ini berhasil merancang dan membangun website billing penyewaan PlayStation berbasis CodeIgniter 4 yang mampu menjawab permasalahan utama pada usaha rental AS2 Lalung, yaitu pencatatan durasi bermain yang masih manual, perhitungan biaya yang tidak akurat, serta kesulitan dalam penyusunan laporan pendapatan. Sistem yang dikembangkan telah melalui serangkaian tahapan sistematis mulai dari analisis kebutuhan, perancangan, implementasi, hingga pengujian, dan secara fungsional terbukti mampu mengotomatisasi seluruh proses bisnis rental mulai dari pencatatan waktu mulai dan selesai, perhitungan biaya dengan pembulatan per 15 menit, hingga penyajian laporan dalam tiga periode waktu. Kontribusi utama dari penelitian ini adalah pengembangan algoritma billing dengan pembulatan waktu per 15 menit yang lebih proporsional dibandingkan sistem per jam konvensional, yang memberikan keadilan tarif bagi pelanggan dengan durasi singkat, serta penyediaan sistem manajemen unit PlayStation yang dinamis yang memungkinkan pengelola menambah, mengedit, menghapus, serta mengaktifkan atau menonaktifkan unit dengan tarif berbeda-beda. Sistem ini juga menghadirkan fitur laporan pendapatan multi-filter (harian, mingguan, bulanan) yang memudahkan pemilik usaha dalam memantau omzet secara real-time, serta implementasi berbasis CodeIgniter 4 yang ringan, cepat, dan mudah di-deploy pada hosting dengan spesifikasi terbatas, sehingga dapat diadopsi oleh rental PlayStation skala kecil menengah lainnya. Keberhasilan sistem ini ditandai dengan beberapa pencapaian signifikan. Pengujian black box terhadap 18 skenario menunjukkan seluruh fitur berfungsi dengan baik tanpa ditemukan kesalahan fatal, sehingga sistem dinyatakan layak digunakan. Dari sisi efisiensi operasional, terjadi peningkatan yang sangat signifikan di mana waktu pemrosesan transaksi berkurang dari rata-rata 2-3 menit menjadi kurang dari 30 detik, yang berarti efisiensi meningkat lebih dari 75%. Selain itu, akurasi perhitungan biaya mencapai 100% selama masa uji coba, mengeliminasi kesalahan hitung yang sebelumnya sering terjadi pada sistem manual terutama untuk durasi yang tidak bulat. Pembuatan laporan pendapatan yang sebelumnya memakan waktu 30-60 menit kini dapat dilakukan dalam hitungan detik, memungkinkan pemilik usaha memantau omzet secara real-time. Sistem ini juga unggul dibandingkan penelitian sejenis karena menerapkan pembulatan waktu per 15 menit yang lebih adil bagi pelanggan, menyediakan fitur manajemen unit PlayStation yang dinamis dengan tarif berbeda-beda, serta laporan pendapatan multi-filter yang tidak ditemukan pada sistem rental PlayStation lainnya. Penggunaan CodeIgniter 4 juga memberikan keunggulan tersendiri karena framework ini ringan, cepat, dan mudah diimplementasikan di lingkungan hosting dengan spesifikasi terbatas. Meskipun demikian, penelitian ini masih memiliki beberapa keterbatasan. Sistem belum mendukung pembayaran non-tunai, tidak memiliki fitur pemesanan online, laporan belum dapat diekspor ke Excel atau PDF, dan belum terintegrasi dengan perangkat keras otomatis. Namun demikian, secara keseluruhan sistem billing ini telah berhasil meningkatkan efisiensi, akurasi, dan ketertiban pengelolaan transaksi penyewaan PlayStation di AS2 Lalung. Dengan demikian, ketiga hipotesis yang diajukan dapat dibuktikan kebenarannya: sistem mampu mencatat durasi dan menghitung biaya secara otomatis, menyimpan data transaksi dan menghasilkan laporan terstruktur, serta meningkatkan efisiensi dan ketertiban pengelolaan transaksi. Penelitian selanjutnya disarankan untuk mengembangkan fitur reservasi online, integrasi dengan payment gateway, ekspor laporan ke berbagai format, serta koneksi dengan perangkat keras untuk mematikan PS secara otomatis ketika waktu habis.

REFERENCES

- [1] R. J. A. Medfora, J. Laoli, C. Y. Waruwu, and A. G. Zai, "Evaluasi Implementasi Framework Codeigniter Pada Pengembangan Aplikasi Web: Kelebihan Dan Kekurangan," *IDENTIK: Jurnal Ilmu Ekonomi, Pendidikan dan Teknik*, vol. 2, no. 1, pp. 82–87, Jan. 2025, doi: 10.70134/identik.v2i1.248.
- [2] P. A. Dzikri, V. Atina, and J. Maulindar, "Pemodelan Sistem Informasi Rental Playstation Berbasis Web Pada Sanjaya Playstation Menggunakan Metode Waterfall," *JEKIN - Jurnal Teknik Informatika*, vol. 4, no. 3, pp. 688–697, Aug. 2024, doi: 10.58794/jekin.v4i3.939.
- [3] A. S. Muharam, Solihin, E. Retnaldi, E. Rahmawati, and L. Emalia, "Website-Based Boarding House Rental Information System (Case Study at Pondok Saluyu Company)," *Management, and Entrepreneurship*, vol. 2, no. 2, pp. 146–163, Dec. 2024, doi: 10.55208/jeme.
- [4] A. N. Kholili and P. Ardiyansyah, "Jurnal Restikom : Riset Teknik Informatika dan Komputer Transformasi Digital Layanan Rental Playstation Menggunakan Aplikasi Mobile Berbasis Design Thinking," *Jurnal Restikom : Riset Teknik Informatika dan Komputer*, vol. 7, no. 1, pp. 137–147, Apr. 2025, doi: 10.52005/restikom.v5i1.
- [5] L. A. Kaka, A. A. Rangga, and E. D. Ege, "Implementasi Framework Code Igniter pada Aplikasi E-Posyandu Berbasis Web," *Router : Jurnal Teknik Informatika dan Terapan*, vol. 3, no. 4, pp. 25–48, Dec. 2025, doi: 10.62951/router.v3i4.714.
- [6] R. Riflansyah and H. Idwan, "Perancangan Sistem Informasi Website Kantor Desa Beurawe Berbasis Framework CodeIgniter," *Jurnal Manajemen Sistem Informasi (JMASIF)*, vol. 4, no. 2, pp. 93–101, Oct. 2025, doi: 10.59431/jmasif.v4i2.605.
- [7] R. Imsar and E. Maiyana, "Sistem Pendaftaran Layanan Internet Di Sekolah Menggunakan Framework Codeigniter 4," *Jurnal Sosial Teknologi*, vol. 4, no. 3, pp. 206–212, Mar. 2024, doi: 10.59188/jurnalsostech.v4i3.1038.
- [8] M. I. Hanafi, A. Triadi, and M. Yusuf, "Rancang Bangun Sistem Informasi Pengajuan Publikasi Berbasis Website Di Badan Pusat Statistik Provinsi Jambi Menggunakan Framework Codeigniter 4 (CI4)," *Jurnal Komputer Teknologi Informasi Sistem Informasi (JUKTISI)*, vol. 4, no. 2, pp. 1017–1026, Aug. 2025, doi: 10.62712/juktisi.v4i2.561.
- [9] E. S. P. Nugroho and F. D. Astuti, "Implementasi PHP Menggunakan Framework CodeIgniter pada Web Pemasaran Kolektif Collaboracion Space," *Jurnal Sosial Teknologi*, vol. 5, no. 3, pp. 584–594, Mar. 2025, doi: 10.59188/jurnalsostech.v5i3.32014.
- [10] I. Burhani, J. Juwari, A. Soderi, and K. Diantoro, "Perbandingan Metodologi SDLC Waterfall dan Agile Dalam Rencana Pengembangan Sistem Informasi Kepatuhan," *Journal of Informatics Management and Information Technology*, vol. 5, no. 2, pp. 147–154, Apr. 2025, doi: 10.47065/jimat.v5i2.489.



- [11] S. Sangaji, S. D. Anggita, and I. R. Wulandari, "Implementasi Metode Waterfall pada Sistem Informasi Laporan Hasil Belajar Siswa Berbasis Website," *IJCSR: The Indonesian Journal of Computer Science Research*, vol. 4, no. 2, pp. 159–168, Jul. 2025, doi: 10.59095/ijcsr.v4i2.203.
- [12] S. Aji, F. Fandhilah, H. Faqih, and R. Rousyati, "Pengembangan Aplikasi Koperasi Simpan Pinjam Menggunakan Metode Waterfall," *JEKIN - Jurnal Teknik Informatika*, vol. 4, no. 2, pp. 88–95, May 2024, doi: 10.58794/jekin.v4i2.706.
- [13] Y. D. D. A. S. Kumara and S. N. Wahyuni, "Perancangan Sistem Informasi Berbasis Web Pada D'lofa Laundry Menggunakan Metode Waterfall," *IJCSR: The Indonesian Journal of Computer Science Research*, vol. 3, no. 1, pp. 10–17, Jan. 2024, doi: 10.59095/ijcsr.v3i1.85.
- [14] D. Dewangga, Y. Wiranata, and L. Y. Hidayat, "Rancang Bangun Sistem E-Library Berbasis Web Menggunakan CodeIgniter 4 di SMP YPI Bintaro," *Jurnal Penelitian Nusantara*, vol. 2, no. 6, pp. 456–463, Jun. 2026, doi: 10.59435/menulis.v2i6.1232.
- [15] N. S. D. Costa, P. K. Karo, and N. M. Faizah, "Perancangan Sistem Informasi Penerimaan Siswa Baru di SMK Harapan Bangsa Berbasis Web Menggunakan CodeIgniter 4," *Computer Journal*, vol. 2, no. 2, pp. 83–90, Aug. 2024, doi: 10.58477/cj.v2i2.172.
- [16] R. Permana, G. Ahmad, K. Gustav, and R. Faiz, "Development of a Web and Mobile-Based E-Business System for Online PlayStation Rental," *Jurnal of Information System and Information Technology*, vol. 2, no. 2, pp. 181–190, Jun. 2025, doi: 10.33197/justinfo.v2i2.2528.
- [17] S. Andhika, D. Yuni, M. Sri, E. Rahmawati, and S. Diantika, "Rancang Bangun Sistem Koperasi Kredit Pada Koperasi Sido Manunggal Berbasis Website Dengan Codeigniter 4," *ANTIVIRUS: Jurnal Ilmiah Teknik Informatika*, vol. 19, no. 2, pp. 264–274, Nov. 2025, doi: 10.35457/antivirus.v19i2.4773.
- [18] I. Ramadhani, R. Nindyasari, and A. C. Murti, "Design and Development of a Web-Based Point of Sale System for Small-Scale Retail Management," *bit-Tech*, vol. 8, no. 1, pp. 181–189, Aug. 2025, doi: 10.32877/bt.v8i1.2487.
- [19] CodeIgniter 4 User Guide, "CodeIgniter Foundation." [Online]. Available: https://codeigniter.com/user_guide/
- [20] N. T. Romadloni, *Sistem Basis Data : Teori dan Praktikum*, vol. 1. PT. Sonpedia Publishing Indonesia, 2024. [Online]. Available: www.sonpedia.com