

RANCANG BANGUN SISTEM INFORMASI PENGELOLAAN KEUANGAN PRIBADI BERBASIS WEB

Moh. Dicky Purnama^{1*}, Aji Primajaya², Adhi Rizal³

^{1,2,3}Universitas Singaperbangsa Karawang; Jl. HS. Ronggo Waluyo, Puseurjaya, Telukjambe Timur, Karawang, Jawa Barat 41361; 0267641177

Keywords:

Sistem Informasi; Keuangan Pribadi; Web Application; Waterfall; *Black-Box Testing*; UAT

Abstrak. Pengelolaan keuangan pribadi yang belum terorganisir dengan baik masih menjadi permasalahan umum, terutama di kalangan mahasiswa dan pekerja muda, yang berdampak pada kesulitan dalam memantau kondisi keuangan secara menyeluruh serta berpotensi menimbulkan pengeluaran yang tidak terkendali. Penelitian ini bertujuan untuk merancang dan mengembangkan sistem informasi pengelolaan keuangan pribadi berbasis web bernama *Control Money* yang dapat membantu pengguna dalam mencatat, mengelola, dan menganalisis keuangan secara lebih efektif. Metode yang digunakan adalah *Software Development Life Cycle* (SDLC) dengan model *Waterfall* yang meliputi tahapan analisis kebutuhan, perancangan sistem, implementasi, pengujian, dan pemeliharaan. Sistem dikembangkan menggunakan teknologi React.js, Node.js, Express.js, dan MySQL. Hasil penelitian menunjukkan bahwa sistem berhasil dibangun dengan fitur utama berupa pencatatan pemasukan dan pengeluaran, pengelolaan dompet, pengaturan anggaran bulanan, serta penyajian laporan keuangan secara real-time. Pengujian menggunakan *Black-Box Testing* menunjukkan bahwa seluruh fungsi sistem berjalan sesuai dengan yang diharapkan, sedangkan hasil *User Acceptance Testing* (UAT) menunjukkan bahwa sistem dapat diterima dan digunakan dengan baik oleh pengguna. Dengan demikian, sistem yang dikembangkan mampu meningkatkan efektivitas pengelolaan keuangan pribadi secara lebih terstruktur.



Copyright © 2026 The Author(s), Published by Jurnal Informatika dan Teknik Elektro Terapan. This is an open-access article distributed under the terms of the Creative Commons Attribution-NonCommercial 4.0 International License (CC BY-NC 4.0).

Abstract. Personal financial management that is not well organized remains a common issue, particularly among students and young workers, leading to difficulties in monitoring overall financial conditions and potentially causing uncontrolled spending. This study aims to design and develop a web-based personal financial management information system called *Control Money*, which assists users in recording, managing, and analyzing their finances more effectively. The method used in this study is the *Software Development Life Cycle* (SDLC) with the *Waterfall* model, consisting of requirement analysis, system design, implementation, testing, and maintenance stages. The system is developed using React.js, Node.js, Express.js, and MySQL. The results show that the system has been successfully implemented with key features such as income and expense recording, wallet management, monthly budgeting, and real-time financial reporting. *Black-Box Testing* results indicate that all system functions operate as expected, while *User Acceptance Testing* (UAT) results show that the system is well accepted and can be effectively used by users. Therefore, the developed system is able to improve the effectiveness of personal financial management in a more structured manner..

1. PENDAHULUAN

Pengelolaan keuangan pribadi menjadi salah satu aspek penting dalam kehidupan modern, khususnya di era digital saat ini. Namun demikian, masih banyak individu yang belum memiliki sistem pencatatan keuangan yang terorganisir dengan baik, sehingga mengalami kesulitan dalam memantau kondisi finansial secara menyeluruh [1]. Rendahnya literasi keuangan juga menjadi salah satu faktor yang mempengaruhi kemampuan individu dalam mengelola keuangan [2].

Seiring dengan perkembangan teknologi informasi, berbagai solusi digital telah dikembangkan untuk membantu pengelolaan keuangan pribadi, terutama dalam bentuk aplikasi mobile. Meskipun memiliki keunggulan dalam performa dan integrasi perangkat, aplikasi mobile memiliki keterbatasan seperti kebutuhan instalasi dan kompatibilitas sistem [3]. Sebagai alternatif, aplikasi berbasis web menawarkan fleksibilitas akses lintas perangkat tanpa instalasi serta kemudahan pemeliharaan sistem secara terpusat [4].

Berdasarkan penelitian terdahulu, sebagian besar sistem pengelolaan keuangan berbasis web masih terbatas pada pencatatan transaksi sederhana dan belum mengintegrasikan fitur pengelolaan anggaran secara komprehensif [5]. Selain itu, evaluasi sistem berbasis pengguna juga masih terbatas sehingga belum sepenuhnya mencerminkan kebutuhan pengguna [6].

Berdasarkan kondisi tersebut, penelitian ini bertujuan untuk mengembangkan sistem informasi pengelolaan keuangan pribadi berbasis web yang mampu membantu pengguna dalam mencatat, mengelola, serta menganalisis kondisi keuangan secara lebih efektif dan terstruktur. Sistem dikembangkan menggunakan metode *Waterfall* dalam kerangka *Software Development Life Cycle* (SDLC) serta diuji menggunakan *Black-Box Testing* dan *User Acceptance Testing* (UAT) untuk memastikan kualitas dan penerimaan sistem.

2. TINJAUAN PUSTAKA

2.1 Sistem Informasi

Sistem informasi merupakan integrasi antara teknologi informasi dan peran manusia yang digunakan untuk mengumpulkan, mengolah, menyimpan, dan menyebarkan informasi sebagai pendukung dalam proses pengambilan keputusan [7]. Pada penelitian ini, sistem informasi diterapkan untuk memfasilitasi individu dalam mengelola keuangan pribadi secara digital.

2.2 Pengelolaan Keuangan Pribadi

Pengelolaan keuangan pribadi adalah proses perencanaan, pencatatan, dan pengendalian keuangan individu yang mencakup pemasukan, pengeluaran, serta perencanaan anggaran [2]. Rendahnya literasi keuangan menyebabkan banyak individu belum mampu mengelola keuangan secara efektif, sehingga diperlukan alat bantu digital untuk mendukung proses tersebut.

2.3 Aplikasi Berbasis Web

Aplikasi berbasis web adalah jenis aplikasi yang dapat diakses melalui browser tanpa memerlukan proses instalasi pada perangkat pengguna [8]. Keunggulan utama aplikasi web adalah fleksibilitas akses lintas platform serta kemudahan dalam pemeliharaan dan pengembangan sistem secara terpusat.

2.4 JavaScript

JavaScript merupakan bahasa pemrograman yang digunakan untuk mengembangkan aplikasi web yang interaktif dan dinamis. Bahasa ini memungkinkan manipulasi elemen antarmuka secara langsung sehingga dapat meningkatkan pengalaman pengguna [9]. Selain itu, JavaScript juga dapat digunakan pada sisi server melalui teknologi Node.js.

2.5 React.js

React.js merupakan library JavaScript yang digunakan untuk membangun antarmuka pengguna berbasis komponen. React mendukung pengembangan aplikasi yang efisien melalui konsep *reusable component* serta penggunaan virtual DOM yang dapat meningkatkan performa aplikasi [10].

2.6 Node.js dan Express.js

Node.js merupakan *runtime environment* yang memungkinkan JavaScript dijalankan pada sisi server dengan pendekatan *asynchronous* dan *event-driven*. Express.js merupakan framework yang digunakan untuk mempermudah pembuatan API dan pengelolaan *routing* dalam pengembangan backend aplikasi web [11].

2.7 MySQL

MySQL adalah sistem manajemen basis data relasional yang digunakan untuk menyimpan serta mengelola data secara terstruktur. MySQL mendukung operasi CRUD serta menjaga integritas data melalui relasi antar tabel [12].

2.8 Software Development Life Cycle (SDLC)

Software Development Life Cycle (SDLC) merupakan kerangka kerja yang digunakan dalam proses pengembangan perangkat lunak, yang meliputi tahapan perencanaan, analisis, perancangan, implementasi, pengujian, hingga pemeliharaan [13]. SDLC membantu memastikan bahwa sistem yang dikembangkan sesuai dengan kebutuhan pengguna.

2.9 Model Waterfall

Model *Waterfall* merupakan salah satu pendekatan dalam SDLC yang bersifat berurutan, di mana setiap tahap harus diselesaikan sebelum melanjutkan ke tahap berikutnya [14]. Model ini mencakup tahapan analisis kebutuhan, perancangan sistem, implementasi, pengujian, serta pemeliharaan.

2.10 Unified Modeling Language (UML)

Unified Modeling Language (UML) merupakan bahasa pemodelan visual yang digunakan untuk merancang serta mendokumentasikan sistem perangkat lunak [15]. UML membantu pengembang dalam memahami struktur dan alur sistem sebelum proses implementasi dilakukan.

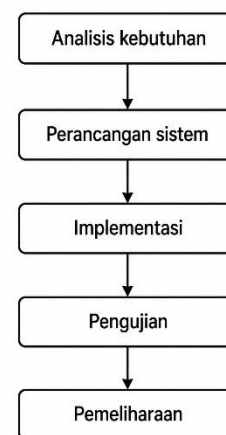
3. METODE PENELITIAN

Penelitian ini menggunakan pendekatan *Software Development Life Cycle* (SDLC) dengan model *Waterfall* sebagai metode pengembangan sistem. Model ini dipilih karena memiliki tahapan yang sistematis dan terstruktur, sehingga memudahkan proses

pengembangan perangkat lunak dari tahap awal hingga akhir [14].

Tahapan dalam model *Waterfall* terdiri dari lima tahap utama, yaitu analisis kebutuhan, perancangan sistem, implementasi, pengujian, dan pemeliharaan. Setiap tahap dilakukan secara berurutan dan menghasilkan output yang menjadi dasar bagi tahap selanjutnya.

Gambar 3.1 *Waterfall* menunjukkan tahapan model *Waterfall* yang digunakan dalam penelitian ini.



Gambar 3. 1 Waterfall

3.1 Analisis Kebutuhan

Tahap analisis kebutuhan dilakukan untuk mengidentifikasi kebutuhan sistem berdasarkan permasalahan yang dihadapi pengguna dalam pengelolaan keuangan pribadi. Pada tahap ini, dilakukan pengumpulan dan analisis data guna mengetahui fitur-fitur yang dibutuhkan, baik dari sisi fungsional maupun non-fungsional.

3.2 Perancangan Sistem

Tahap perancangan sistem dilakukan sebagai lanjutan dari hasil analisis kebutuhan. Pada tahap ini, dilakukan penyusunan desain sistem yang mencakup arsitektur aplikasi, perancangan basis data, serta desain antarmuka pengguna.

Perancangan sistem dilakukan dengan menggunakan Unified Modeling Language (UML), seperti use case diagram untuk menggambarkan interaksi antara pengguna dengan sistem, serta diagram lainnya untuk memodelkan alur proses dan struktur sistem. Hasil dari tahap ini menjadi acuan dalam proses

pengembangan sistem pada tahap berikutnya [15].

3.3 Implementasi

Tahap implementasi merupakan proses pengembangan sistem berdasarkan hasil perancangan yang telah dibuat sebelumnya. Pada tahap ini, seluruh komponen sistem mulai dibangun dan diintegrasikan menjadi sebuah aplikasi yang dapat digunakan.

Sistem dikembangkan menggunakan teknologi berbasis web dengan memanfaatkan React.js sebagai frontend, serta Node.js dan Express.js sebagai backend. Selain itu, MySQL digunakan sebagai basis data untuk menyimpan dan mengelola data pengguna. Hasil dari tahap ini adalah sistem yang siap untuk diuji pada tahap selanjutnya.

3.4 Pengujian

Pengujian yang bertujuan untuk memastikan bahwa sistem berjalan sesuai dengan kebutuhan. Pengujian dilakukan menggunakan metode *Black-Box Testing* untuk menguji fungsi sistem [16], serta *User Acceptance Testing* (UAT) untuk mengetahui tingkat penerimaan pengguna terhadap sistem [17].

3.5 Pemeliharaan

Pemeliharaan yaitu proses perbaikan dan pengembangan sistem setelah sistem digunakan, guna menjaga kinerja dan kualitas sistem agar tetap optimal.

4. HASIL DAN PEMBAHASAN

4.1 Analisis Kebutuhan

Analisis kebutuhan dilakukan untuk merumuskan kebutuhan sistem berdasarkan permasalahan yang dialami pengguna dalam pengelolaan keuangan pribadi. Data dikumpulkan melalui penyebaran kuesioner kepada responden yang berasal dari kalangan mahasiswa dan pekerja, sehingga diperoleh gambaran mengenai kebiasaan serta hambatan dalam mengelola keuangan sehari-hari.

Berdasarkan hasil analisis, pengguna membutuhkan sistem yang mampu memfasilitasi pencatatan pemasukan dan pengeluaran, pengelolaan dompet, serta penyajian data keuangan secara terorganisir dan real-time.

Adapun kebutuhan fungsional sistem meliputi fitur pencatatan transaksi, pengelompokan kategori, pengaturan anggaran, dan penyajian laporan keuangan. Sedangkan kebutuhan non-fungsional mencakup aspek kemudahan penggunaan, aksesibilitas sistem berbasis web, serta performa yang stabil dan responsif.

4.2 Perancangan Sistem

Tahap perancangan sistem menggunakan Unified Modeling Language (UML) yang terdiri dari use case diagram, activity diagram, sequence diagram, dan class diagram [15]. se case diagram digunakan untuk menunjukkan hubungan interaksi antara pengguna dan sistem, sementara activity diagram digunakan untuk memodelkan alur aktivitas atau proses yang berlangsung dalam sistem.

Sequence diagram menggambarkan alur komunikasi antar komponen sistem, sementara class diagram digunakan untuk mendeskripsikan struktur data dan relasi antar entitas dalam sistem. Perancangan ini menjadi dasar dalam proses implementasi sistem.

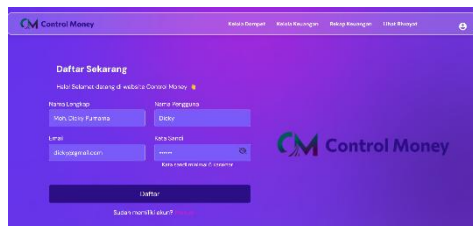
4.3 Implementasi

Pengembangan sistem informasi pengelolaan keuangan pribadi dilakukan dalam bentuk aplikasi berbasis web dengan menggunakan React.js sebagai antarmuka pengguna, serta Node.js dan Express.js sebagai pengelola sisi server. Data sistem disimpan menggunakan MySQL sebagai basis data. Pada tahap implementasi, sistem menghasilkan beberapa fitur utama, antara lain:

1. Halaman daftar berfungsi sebagai gerbang awal bagi pengguna untuk melakukan proses autentikasi sebelum mengakses sistem. Melalui halaman ini, pengguna dapat melakukan pendaftaran akun agar dapat masuk ke dalam aplikasi.

Selain itu, sistem juga menerapkan validasi pada setiap input yang diberikan pengguna untuk memastikan kesesuaian data dengan aturan yang berlaku. Penerapan ini bertujuan untuk meningkatkan tingkat keamanan serta menjaga konsistensi data yang tersimpan.

Tampilan halaman login dan registrasi dapat dilihat pada Gambar 4.1 Daftar.

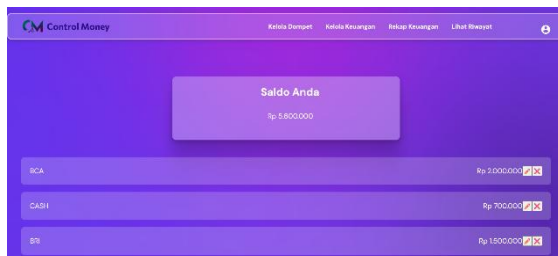


Gambar 4. 1 Daftar

2. Halaman kelola dompet digunakan untuk mengatur sumber keuangan pengguna dalam sistem. Pada halaman ini, pengguna diberikan kemampuan untuk menambah, mengubah, serta menghapus data dompet sesuai dengan kondisi yang diinginkan.

Fitur ini memungkinkan pengguna untuk memisahkan sumber keuangan sehingga pencatatan transaksi menjadi lebih terorganisir dan mudah dipantau.

Tampilan halaman kelola dompet dapat dilihat pada Gambar 4.2 Kelola Dompet.

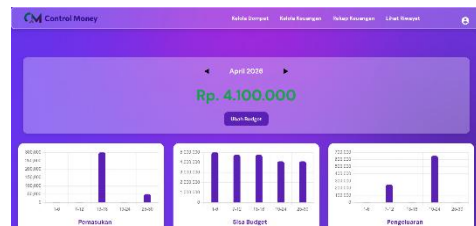


Gambar 4. 2 Kelola Dompet

3. Halaman kelola keuangan merupakan fitur inti dalam sistem yang digunakan untuk mencatat transaksi pemasukan dan pengeluaran. Pada halaman ini, pengguna dapat menambahkan data transaksi yang dilengkapi dengan kategori, nominal, serta tanggal transaksi.

Melalui fitur ini, pengguna dapat melakukan pencatatan keuangan secara lebih terorganisir sehingga memudahkan dalam memantau dan mengevaluasi kondisi keuangan.

Tampilan halaman kelola keuangan dapat dilihat pada Gambar 4.3 Kelola Keuangan.



Gambar 4. 3 Kelola Keuangan

4. Halaman rekap keuangan digunakan untuk menampilkan ringkasan kondisi keuangan pengguna secara otomatis. Pada halaman ini, sistem menyajikan informasi total pemasukan, pengeluaran, serta sisa saldo dalam periode tertentu.

Selain itu, data disajikan dalam bentuk visualisasi guna membantu pengguna memahami kondisi keuangan secara lebih cepat dan akurat.

Tampilan halaman rekap keuangan dapat dilihat pada Gambar 4.4 Rekap Keuangan.

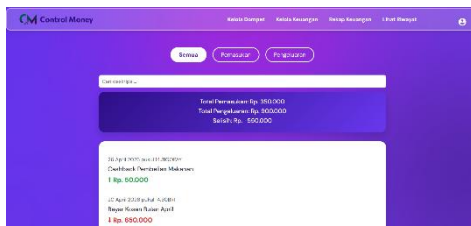


Gambar 4. 4 Rekap Keuangan

5. Halaman riwayat transaksi berfungsi untuk menampilkan seluruh catatan transaksi yang telah dilakukan oleh pengguna. Pada halaman ini, pengguna dapat melihat rincian pemasukan dan pengeluaran secara berurutan berdasarkan waktu.

Fitur ini membantu pengguna dalam menelusuri kembali data transaksi serta melakukan evaluasi terhadap pola pengeluaran yang dimiliki.

Tampilan halaman riwayat transaksi dapat dilihat pada Gambar 4.5 Riwayat Transaksi.

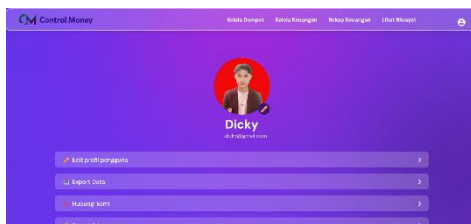


Gambar 4. 5 Riwayat Transaksi

6. Halaman profil berfungsi untuk mengelola informasi pengguna dalam sistem. Melalui halaman ini, pengguna dapat melihat serta memperbarui data pribadi yang telah terdaftar.

Fitur ini mendukung pengelolaan akun agar tetap relevan dan sesuai dengan identitas pengguna.

Tampilan halaman profil dapat dilihat pada Gambar 4.6 Profil.



Gambar 4. 6 Profil

Berdasarkan hasil implementasi, seluruh fitur utama sistem telah berjalan dengan baik dan sesuai dengan kebutuhan yang telah diidentifikasi pada tahap analisis.

4.4 Pengujian

Pengujian sistem dilakukan untuk memastikan bahwa aplikasi yang dikembangkan berfungsi sesuai dengan kebutuhan serta dapat diterima oleh pengguna. Proses pengujian dilakukan menggunakan metode *Black-Box Testing* dan *User Acceptance Testing* (UAT).

4.4.1 Black-Box Testing

Metode *Black-Box Testing* digunakan untuk menguji fungsionalitas sistem tanpa melihat implementasi kode di dalamnya [16]. Pengujian ini bertujuan untuk memastikan bahwa seluruh

fitur berjalan sesuai dengan spesifikasi yang telah ditentukan.

Tabel 4.1 *Black-Box Testing* menunjukkan hasil pengujian *Black-Box Testing* pada sistem yang dikembangkan.

Tabel 4. 1 Black-Box Testing

No	Pengujian	Skenario Pengujian	Hasil yang Diharapkan	Hasil Aktual	Status
1	Login	Pengguna memasukkan email dan password yang valid	Sistem berhasil mengarahkan pengguna ke halaman dashboard	Berhasil masuk ke dashboard	Lulus
2	Register	Pengguna mengisi data registrasi dengan benar	Akun baru berhasil dibuat dan dapat digunakan untuk login	Akun berhasil dibuat	Lulus
3	Kelola Dompet	Pengguna menambahkan data dompet baru	Data dompet tersimpan dan ditampilkan pada daftar dompet	Data berhasil ditambahkan	Lulus
4	Kelola Keuangan (Atur Budget)	Pengguna mengatur anggaran/budget bulanan	Data budget tersimpan dan sistem menampilkan sisa anggaran	Data berhasil disimpan	Lulus
5	Kelola Keuangan	Pengguna menambahkan transaksi pemasukan	Data pemasukan tersimpan dan saldo bertambah	Data berhasil disimpan	Lulus
6	Kelola Keuangan	Pengguna menambahkan transaksi pengeluaran	Data pengeluaran tersimpan dan sisa budget bulanan berkurang	Data berhasil disimpan	Lulus
7	Rekap Keuangan	Pengguna memilih periode rekap (harian/bulanan/tahunan)	Sistem menampilkan data rekap sesuai periode	Data rekap tampil sesuai	Lulus
8	Lihat Riwayat	Pengguna membuka halaman riwayat transaksi	Sistem menampilkan seluruh data transaksi	Data riwayat tampil	Lulus
9	Profil	Pengguna mengubah data profil	Data profil berhasil diperbarui	Data berhasil diperbarui	Lulus

4.4.2 User Acceptance Testing (UAT)

Pengujian *User Acceptance Testing* (UAT) bertujuan untuk memastikan bahwa sistem yang telah dikembangkan mampu memenuhi kebutuhan pengguna dan dapat dioperasikan dengan baik dalam situasi sebenarnya [18].

Pengujian dilakukan dengan melibatkan 5 responden yang diminta untuk mencoba langsung fitur utama pada aplikasi *Control Money* berdasarkan skenario yang telah ditentukan.

Setiap responden melakukan pengujian terhadap beberapa fitur utama, seperti login, pengelolaan dompet, pencatatan keuangan, rekap keuangan, riwayat transaksi, serta pengelolaan profil. Hasil pengujian dicatat dalam bentuk status “Lulus” atau “Tidak Lulus” berdasarkan kesesuaian antara hasil aktual dengan hasil yang diharapkan.

Tabel 4. 2 User Acceptance Testing

No	Fitur	Skenario Pengujian	Jumlah Responden	Lulus	Tidak Lulus
1	Login/Regi ster	Login dengan data valid	5	5	0
2	Kelola Dompet	Tambah, ubah, hapus dompet	5	5	0
3	Kelola keuangan	set budget & input transaksi	5	5	0
4	Rekap Keuangan	Menampil kan rekap multi rentang waktu	5	5	0
5	Lihat Riwayat	Melihat Riwayat	5	5	0
6	Profil	Update profil, export data, privasi & keamanan	5	5	0

Berdasarkan hasil UAT yang melibatkan 5 responden, seluruh fitur utama dalam sistem memperoleh status “Lulus”. Hal ini menunjukkan bahwa sistem telah berfungsi sesuai dengan skenario yang ditetapkan serta mampu memenuhi kebutuhan pengguna.

Selain itu, hasil pengujian juga menunjukkan bahwa sistem mudah digunakan dan dapat dioperasikan tanpa kendala yang berarti. Dengan demikian, sistem informasi pengelolaan keuangan pribadi yang dikembangkan dapat diterima dengan baik dan layak digunakan dalam aktivitas pengelolaan keuangan sehari-hari.

4.5 Pemeliharaan

Tahap pemeliharaan dilakukan setelah sistem diimplementasikan dan digunakan oleh pengguna. Pemeliharaan bertujuan untuk menjaga kinerja sistem agar tetap optimal serta melakukan perbaikan apabila ditemukan kesalahan (*bug*) selama penggunaan sistem.

Pada penelitian ini, pemeliharaan sistem dilakukan dengan cara melakukan monitoring terhadap kinerja aplikasi, memperbaiki kesalahan yang ditemukan, serta melakukan penyesuaian terhadap kebutuhan pengguna yang mungkin berkembang di masa mendatang. Selain itu, pemeliharaan juga mencakup pembaruan sistem untuk meningkatkan keamanan, performa, dan kompatibilitas dengan teknologi yang digunakan.

Dengan adanya tahap pemeliharaan, sistem yang dikembangkan diharapkan dapat terus digunakan secara berkelanjutan dan mampu beradaptasi dengan kebutuhan pengguna.

4.6 Pembahasan

Berdasarkan hasil implementasi dan pengujian yang telah dilakukan, sistem informasi pengelolaan keuangan pribadi berbasis web yang dikembangkan mampu memenuhi kebutuhan pengguna dalam mencatat serta mengelola keuangan secara terorganisir. Seluruh fitur utama, seperti pencatatan pemasukan dan pengeluaran, pengelolaan dompet, pengaturan anggaran, serta penyajian rekap keuangan, telah berfungsi dengan baik sesuai dengan kebutuhan yang telah diidentifikasi sebelumnya.

Hasil pengujian *Black-Box Testing* menunjukkan bahwa seluruh fungsi sistem berjalan sesuai dengan skenario yang telah ditentukan [16]. Selain itu, hasil *User Acceptance Testing* (UAT) yang melibatkan 5 responden menunjukkan bahwa seluruh fitur sistem memperoleh status “Lulus”, yang menandakan bahwa sistem dapat digunakan dengan baik tanpa kendala yang berarti [17].

Dibandingkan dengan penelitian sebelumnya, sistem yang dikembangkan dalam penelitian ini memiliki keunggulan pada integrasi fitur yang lebih komprehensif, tidak hanya berfokus pada pencatatan transaksi,

tetapi juga mencakup pengelolaan anggaran serta penyajian laporan keuangan secara *real-time* [4]. Hal ini memberikan kemudahan bagi pengguna dalam memantau kondisi keuangan secara menyeluruh.

Di sisi lain, penggunaan teknologi berbasis web memungkinkan pengguna untuk mengakses sistem tanpa harus melakukan instalasi, sehingga dapat digunakan dengan mudah di berbagai perangkat [4]. Hal ini menunjukkan bahwa sistem tidak hanya memenuhi aspek fungsional, tetapi juga memberikan kemudahan dalam akses serta meningkatkan kenyamanan pengguna.

Secara keseluruhan, sistem informasi yang dikembangkan telah mampu mengatasi permasalahan yang diangkat dalam penelitian, yaitu membantu pengguna dalam mengelola keuangan pribadi secara lebih efektif dan terstruktur.

5. KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan, diperoleh beberapa kesimpulan sebagai berikut:

1. Sistem informasi pengelolaan keuangan pribadi berbasis web berhasil dibangun menggunakan metode *Software Development Life Cycle* (SDLC) dengan pendekatan model *Waterfall*.
2. Sistem yang dikembangkan menyediakan fitur utama, seperti pencatatan pemasukan dan pengeluaran, pengelolaan dompet, pengaturan anggaran, serta penyajian rekap keuangan secara *real-time*.
3. Pengujian menggunakan *Black-Box Testing* menunjukkan bahwa seluruh fungsi sistem berjalan sesuai dengan kebutuhan yang telah ditentukan.
4. Hasil *User Acceptance Testing* (UAT) yang melibatkan 5 responden menunjukkan bahwa seluruh fitur sistem dinyatakan "Lulus", sehingga sistem dapat diterima dan digunakan tanpa kendala yang berarti.
5. Sistem yang dikembangkan mampu memberikan kemudahan bagi pengguna dalam

mengelola keuangan pribadi secara lebih efektif, terstruktur, dan mudah diakses melalui platform berbasis web.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih yang sebesar-besarnya kepada semua pihak yang telah memberikan dukungan, baik secara langsung maupun tidak langsung, sehingga penelitian ini dapat terselesaikan dengan baik..

DAFTAR PUSTAKA

- [1] Mustika, N. Yusuf, and V. Taruh, "Pengaruh Literasi Keuangan, Sikap Keuangan Dan Kemampuan Akademik Terhadap Perilaku Pengelolaan Keuangan Mahasiswa Akuntansi Fakultas Ekonomi Universitas Negeri Gorontalo," *J. Mhs. Akunt.*, vol. 1, no. 1, pp. 82–96, Jun. 2022, doi: 10.37479/jamak.v1i1.20.
- [2] N. N. Aisa and F. H. Silalahi, "Analisis Literasi Keuangan Gen-Z: Studi Kasus pada Mahasiswa Akuntansi," *J. Bus. Halal Ind.*, vol. 1, no. 4, Aug. 2024, doi: 10.47134/jbhi.v1i4.385.
- [3] M. Kothapalli, "Progressive web apps: Merging web accessibility with native performance," *Eur. J. Adv. Eng. Technol.*, vol. 2021, no. 4, pp. 60–68, 2021.
- [4] Muhammad Wildan Jaffar Rahmatullah, Nanda Nur Oktavia, and Lucas Lawrence, "Web-Based Personal Finance Management Application with Interactive Data Visualization," *Technomedia J.*, vol. 10, no. 1, pp. 67–78, 2025.
- [5] A. Rosidi and A. Afriyudi, "Aplikasi Pencatatan Keuangan Pribadi Berbasis Web Mobile," *J. Teknol. Inform. dan Komput.*, vol. 9, no. 1, pp. 100–113, Mar. 2023, doi: 10.37012/jtik.v9i1.1447.
- [6] R. Setyaningsih and M. Budiantara, "Penerapan Pencatatan Laporan Keuangan Sederhana pada Toko Rohani Ibu Margi Klaten," *J. Abdi Masy. Indones.*, vol. 3, no. 3, pp. 953–958, May 2023, doi: 10.54082/jamsi.764.
- [7] A. Albert and . E., "Sistem Informasi Kerja Praktik Berbasis Website Untuk Optimasi Program Kerja Praktik Pada Perpustakaan Nasional Republik Indonesia," *J. Sist. Inf. dan Apl.*, vol. 1, no. 1, pp. 80–93, Sep. 2023, doi: 10.52958/jsia.v1i1.5909.
- [8] M. Marlina, M. Masnur, and Muh. Dirga.F, "Aplikasi E-Learning Siswa Smk Berbasis Web," *J. Sintaks Log.*, vol. 1, no. 1, pp. 8–17, Jan. 2021, doi: 10.31850/jsilog.v1i1.672.
- [9] Octa Selsa Is Anggraeni, Lilik Sugiarto, and

- Tinuk Agustin, "Studi Komparatif Performa Framework Javascript Modern dalam Pengembangan Aplikasi Web," *Modem J. Inform. dan Sains Teknol.*, vol. 2, no. 4, pp. 162–177, Sep. 2024, doi: 10.62951/modem.v2i4.239.
- [10] M. de Sousa and A. Goncalves, "humanportal – A React.js case study," in *2020 15th Iberian Conference on Information Systems and Technologies (CISTI)*, IEEE, Jun. 2020, pp. 1–6. doi: 10.23919/CISTI49556.2020.9141070.
- [11] K. Saundariya, M. Abirami, K. R. Senthil, D. Prabakaran, B. Srimathi, and G. Nagarajan, "Webapp Service for Booking Handyman Using MongoDB, Express JS, React JS, Node JS," in *2021 3rd International Conference on Signal Processing and Communication (ICSPC)*, IEEE, May 2021, pp. 180–183. doi: 10.1109/ICSPC51351.2021.9451783.
- [12] Z. Olta, B. Morina, and E. Haskuka, "Optimizing Data Retrieval and Update Operations in PHP-MySQL Applications: A Review," *J. Mod. Technol.*, vol. 02, no. 02, pp. 312–316, Jul. 2025, doi: 10.71426/jmt.v2.i2.pp312-316.
- [13] P. Leloudas, "Software Development Life Cycle," in *Introduction to Software Testing*, Berkeley, CA: Apress, 2023, pp. 35–55. doi: 10.1007/978-1-4842-9514-4_3.
- [14] T. Pricillia and Zulfachmi, "Perbandingan Metode Pengembangan Perangkat Lunak (Waterfall, Prototype, RAD)," *J. Bangkit Indones.*, vol. 10, no. 1, pp. 6–12, Mar. 2021, doi: 10.52771/bangkitindonesia.v10i1.153.
- [15] L. D. Fitrani and A. C. Puspitaningrum, "Utilization Of UML (Unified Modeling Language) In The Design Of Academic Information Systems Based On The OOAD Method," *SISTEMASI*, vol. 12, no. 2, p. 614, May 2023, doi: 10.32520/stmsi.v12i2.2871.
- [16] E. Viglianisi, M. Dallago, and M. Ceccato, "RESTTESTGEN: Automated Black-Box Testing of RESTful APIs," in *2020 IEEE 13th International Conference on Software Testing, Validation and Verification (ICST)*, IEEE, Oct. 2020, pp. 142–152. doi: 10.1109/ICST46399.2020.00024.
- [17] H. Yakub, B. Daniawan, A. Wijaya, and L. Damayanti, "Sistem Informasi E-Commerce Berbasis Website Dengan Metode Pengujian User Acceptance Testing," *JSITIK J. Sist. Inf. dan Teknol. Inf. Komput.*, vol. 2, no. 2, pp. 113–127, Apr. 2024, doi: 10.53624/jsitik.v2i2.362.
- [18] I. G. Ngurah, E. Nandana, and I. G. Arta, "Analisis UX pada Aplikasi Pemutar Video Animasi dengan Metode Usability Testing," vol. 3, pp. 729–736, 2025.