Penerapan Integrasi Model Waterfall dan User-Centered Design

Sri Ratna Wulan[1], Zahwa Putri Hamida[2], Anna Kurniaty[3], Muhammad Riza Alifi[4]
[1,2,3,4]Jurusan Teknik Komputer dan Informatika, Politeknik Negeri Bandung
Email: [1]sri.ratna@polban.ac.id, [2]zahwa.putri.tif420@polban.ac.id, [3]anna.kurniaty.tif420@polban.ac.id, [4]muhammad.riza@polban.ac.id

Abstrak. *User-Centered Design (UCD) adalah pendekatan penting dalam pengembangan perangkat lunak untuk memastikan produk yang dihasilkan ramah pengguna (user-friendly). Namun, penerapan UCD dalam siklus pengembangan perangkat lunak masih jarang dibahas. Penelitian ini bertujuan untuk menguji apakah UCD dapat diintegrasikan ke dalam model Waterfall melalui pengembangan sebuah aplikasi berbasis web. Proses integrasi dilakukan dengan menerapkan UCD pada fase identifikasi kebutuhan, yang kemudian dilanjutkan dengan fase-fase lain, seperti analisis, desain, implementasi, pengujian, dan deployment. Metode penelitian yang digunakan mencakup observasi dan wawancara dengan pelaksana proyek pengembangan aplikasi. Hasil penelitian menunjukkan bahwa integrasi UCD ke dalam model Waterfall sulit dilakukan dikarenakan proses UCD memiliki iterasi yang bisa memakan waktu lebih banyak dari perkiraan. Pada proyek ini, proses UCD pada fase identifikasi kebutuhan dilaksanakan 2 iterasi, namun dikarenakan waktu proyek yang sudah tersita lebih dari perkiraan, ada 1 tahap yakni evaluasi desain yang terlewat. Hal ini berdampak kepada waktu implementasi yang terpotong sehingga menyebabkan fitur yang belum terimplementasi sebanyak 13%. Hasil Unit Testing dan UAT menyatakan produk yang dihasilkan bekerja dengan baik 100%. Namun, dari hasil wawancara terhadap pengguna, mereka mengemukakan produk kurang nyaman digunakan dan kurang estetik sesuai dengan hasil evaluasi heuristik pada proses UCD pada iterasi pertama. Evaluasi heuristik pada iterasi kedua tidak dilakukan kembali karena waktu yang terbatas. Sehingga dapat disimpulkan, iterasi pada proses UCD dapat memakan waktu lebih banyak dari perkiraan sehingga berisiko bertambah lamanya proyek pengembangan dilaksanakan. Di sisi lain, jika proses UCD tidak dikerjakan dengan baik berisiko mengakibatkan produk yang dihasilkan kurang nyaman dan sesuai dengan kebutuhan pengguna.*
Kata Kunci: *user-centered design, ucd, waterfall, integrasi, pengembangan*

Abstract. *User-Centered Design (UCD) is crucial for developing user-friendly software, yet its integration into traditional development cycles like Waterfall is rarely discussed. This study aims to explore the integration of UCD into the Waterfall model through the development of a web-based application. UCD was applied during the requirement identification phase, followed by analysis, design, implementation, testing, and deployment. The research involved observations and interviews with project developers. Results showed that integrating UCD into the Waterfall model is challenging due to the iterative nature of UCD, which often takes more time than planned. In this project, two iterations were conducted during the requirement identification phase, but a key stage—design evaluation—was missed due to time constraints. As a result, 13% of the features remained unimplemented. Unit Testing and UAT confirmed that the product worked well, but user feedback revealed discomfort and dissatisfaction with the design. A second heuristic evaluation was not performed due to time limitations. The study concludes that UCD iterations may extend project timelines, and if not fully executed, can result in a product that does not meet user needs or expectations.*
Keywords: *user-centered design, ucd, waterfall, integration, development*

1. Pendahuluan

Hal yang membedakan antara produk perangkat lunak yang populer digunakan banyak pengguna dengan yang tidak populer adalah kebergunaan (*usability*) dan kenyamanan produk (*user-experience*) tersebut digunakan [1]. Salah satu pendekatan untuk mengembangkan sebuah produk yang memiliki kebergunaan dan kenyamanan produk dapat menggunakan pendekatan berpusat pada pengguna atau yang disebut dengan User-Centered Design (UCD). User-Centered Design (UCD) merupakan sebuah pendekatan desain yang melibatkan pengguna untuk meningkatkan pemahaman mengenai kebutuhan pengguna dan kebutuhan bisnis, dan iterasi

 ISSN: 2579-566X (Online)
ISSN: 2477-5274 (Print)

desain dan evaluasi [2]. Pentingnya UCD dilibatkan dalam pengembangan produk sudah mulai disadari. Namun, yang menjadi masalah adalah jarang ada yang membahas bagaimana UCD diterapkan dalam siklus pengembangan perangkat lunak. Penelitian dengan topik UCD biasanya berupa implementasi UCD dan dilakukan hanya sampai pada tahap perancangan antarmuka dengan output desain antarmuka yang telah dievaluasi. Seperti pada penelitian yang bertujuan untuk mendesain ulang antarmuka [3], [4], [5] atau pun mendesain antarmuka dari awal [6], [7], [8], [9], [10], [11]. Ada beberapa penelitian yang mengembangkan aplikasi dengan metode Waterfall yang diintegrasikan dengan UCD, namun cara pengintegrasiannya kurang jelas [12], [13], [14].

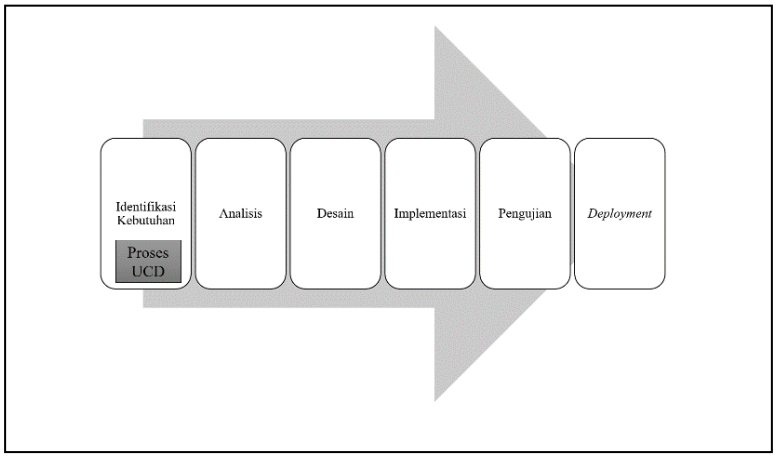
Dhandapani [15] mengusulkan pendekatan untuk mengintegrasikan antara metodologi pengembangan perangkat lunak dengan UCD. Penelitian tersebut mencoba mengintegrasikan berbagai metodologi seperti model Waterfall, Increment, Agile, dan Prototipe dengan UCD. Namun, dalam penelitian tersebut, usulan tersebut belum diimplementasikan ke dalam proyek pengembangan aplikasi nyata sehingga belum dapat dibuktikan apakah model integrasi dapat diterapkan. Pada paper ini, peneliti mencoba mengimplementasikan salah satu integrasi yang diusulkan oleh Dhandapani [15] ke dalam proyek nyata, yakni integrasi antara model Waterfall dan UCD. Studi kasus yang digunakan adalah pengembangan aplikasi pengelolaan indekos. Penelitian ini bertujuan untuk mengetahui apakah pendekatan yang diusulkan Dhandapani [15] untuk mengintegrasikan model Waterfall dan UCD dapat diterapkan ke dalam proyek pengembangan perangkat lunak nyata.

2. Metode Penelitian

Studi kasus yang dipilih untuk menguji bagaimana penerapan integrasi antara model Waterfall dan pendekatan UCD adalah pengembangan aplikasi pengelolaan indekos. Pengguna yang dipilih adalah pemilik atau pengelola indekos di sekitar kampus Politeknik Negeri Bandung. Penelitian dilakukan pada Bulan Januari sampai Juli 2024. Sub bab berikut akan menjelaskan mengenai metode pengembangan aplikasi yang dijadikan studi kasus ini dan metode analisis dilakukan untuk mengetahui bagaimana penerapan tersebut dilakukan.

2.1 Metode Pengembangan Aplikasi

Metode pengembangan aplikasi yang digunakan adalah pendekatan yang diusulkan pada penelitian Dhandapani [15] yang dapat dilihat pada Gambar 1, yakni dengan mengintegrasikan model Waterfall dengan UCD. Integrasi ini dilakukan pada tahap analisis kebutuhan. Dhandapani [15] berargumen bahwa UCD cocok diintegrasikan pada model ini karena masing-masing model membutuhkan pendetailan, terutama pada tahap identifikasi kebutuhan. Jika kebutuhan pengguna bisa diidentifikasi bersamaan dengan kebutuhan bisnis, maka akan tercipta desain yang lebih akurat. Setelah dilakukan identifikasi kebutuhan, maka selanjutnya yang dilakukan adalah tahap analisis, desain, implementasi, pengujian, dan *deployment*.

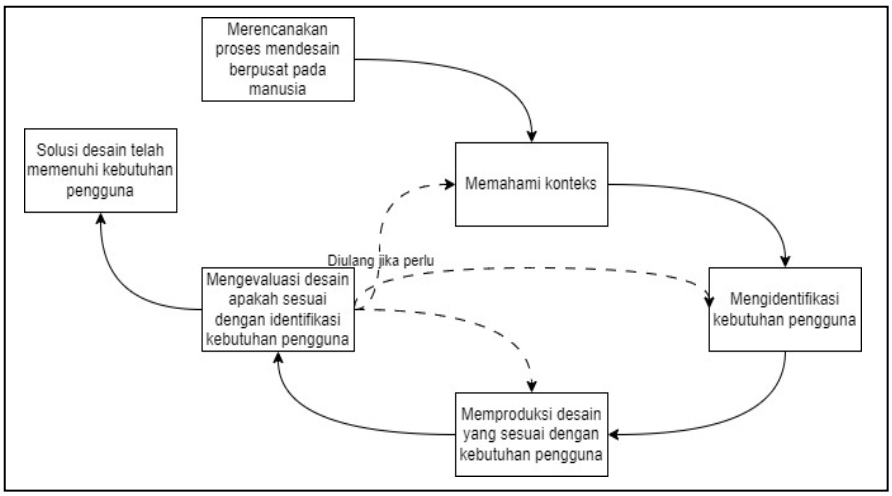


Gambar 1. Integrasi Model Waterfall dan UCD [15]

Fase pertama adalah mengidentifikasi kebutuhan dan berjalannya proses UCD. Fase ini bertujuan untuk memahami pasar dan mengidentifikasi area mana yang dapat diimprovisasi dengan

pengembangan aplikasi yang dilakukan. Pada tahapan ini, identifikasi kebutuhan perangkat lunak dilakukan dengan menerapkan UCD. Tahapan UCD yang dilakukan berdasarkan ISO 9241-210:2010 [16] mengenai desain berpusat pada manusia untuk sistem yang interaktif. Tahapan ini dapat dilihat pada Gambar 2. Tahapan ini diantaranya merencanakan proses desain berpusat pada manusia, memahami konteks, menentukan kebutuhan pengguna dan organisasi, membuat solusi desain, mengevaluasi desain dengan mencocokkan dengan kebutuhan pengguna, dan menghasilkan solusi desain yang sesuai dengan kebutuhan pengguna. Jika desain ternyata tidak sesuai dengan kebutuhan pengguna, maka selanjutnya proses akan kembali lagi ke proses memahami konteks, menentukan kebutuhan pengguna dan organisasi, atau proses membuat solusi desain sesuai dengan hasil evaluasi, apa yang menjadi masalah.

Fase identifikasi kebutuhan ini memiliki 7 tahapan, diantaranya sebagai berikut: (a) merencanakan proses mendesain berpusat pada manusia; proses ini meliputi perencanaan survei untuk memahami konteks masalah, perencanaan wawancara untuk mengumpulkan kebutuhan pengguna, pembuatan instrumen pertanyaan survei dan wawancara, validasi instrumen, dan perencanaan pengolahan hasil wawancara dan survei. (b) memahami konteks; proses ini meliputi kegiatan survei dan wawancara terhadap pengguna dan pengolahan hasil wawancara dan survey. (c) mengidentifikasi kebutuhan pengguna; proses ini meliputi pembuatan user persona, menyimpulkan kebutuhan pengguna, dan kesulitan pengguna berdasarkan hasil olah data wawancara dan survey. (d) Mengumpulkan data kebutuhan pengguna; proses ini meliputi pendataan kebutuhan pengguna. (e) Memproduksi desain sesuai dengan kebutuhan pengguna; proses ini merupakan proses pembuatan desain yang berfokus pada kebutuhan pengguna. Hasil dari proses ini berupa *low-fidelity prototype* untuk iterasi pertama, dan *high fidelity prototype* untuk iterasi selanjutnya. (f) Mengevaluasi desain sesuai dengan kebutuhan pengguna; proses ini meliputi pertemuan dengan pengguna untuk mendapatkan umpan balik mengenai desain, uji heuristik prototipe yang bersifat *high-fidelity*. Hasil dari proses ini berupa kesimpulan apakah desain sudah sesuai dengan kebutuhan pengguna atau belum. Jika belum, maka proses kembali ke langkah (b, c, d, atau e) sesuai hasil evaluasi. (g) Desain memenuhi kebutuhan pengguna; proses ini merupakan proses akhir dimana desain sudah fiks, tidak akan diubah kembali.



Gambar 2. Proses UCD berdasarkan ISO 9241-210:2010 [16]

Fase berikutnya adalah analisis. Fase ini merupakan fase kedua dari model Waterfall. Input untuk fase ini adalah data kebutuhan pengguna yang dikumpulkan pada fase identifikasi kebutuhan. Data dianalisis untuk didetailkan apa saja yang dibutuhkan untuk mengembangkan perangkat lunak yakni cakupan proyek, kebutuhan fungsional, kebutuhan antarmuka eksternal, dan kebutuhan teknikal. Hasil dari fase ini adalah dokumen *Software Requirement Specification* (SRS). Fase selanjutnya adalah desain. Fase ini merupakan fase ketiga dari model Waterfall untuk mendesain perangkat lunak mencakup proses mendesain arsitektur dan antarmuka. Namun, dikarenakan tahap desain antarmuka telah dilakukan pada fase pertama yakni identifikasi kebutuhan, maka output pada fase ini hanya desain

arsitektur perangkat lunak. Fase selanjutnya adalah implementasi. Fase ini merupakan fase keempat dari model Waterfall di mana desain yang sudah dibuat diimplementasikan ke dalam kode pemrograman sehingga menghasilkan produk aplikasi yang diinginkan. Fase ini mencakup pembuatan basis data dan pemrograman. Fase selanjutnya adalah pengujian. Fase ini merupakan fase kelima dari model Waterfall di mana produk diuji. Pengujian yang dilakukan berupa Black-Box Testing dan User Testing. Jika ada kesalahan, maka produk diperbaiki namun sebatas kesalahan yang berasal dari tahap implementasi bukan dari tahapan identifikasi kebutuhan, analisis, atau desain. Fase terakhir adalah *deployment.* Fase terakhir adalah *deployment* yang mana hasil produk disimpan ke dalam server dan dibuat agar produk dapat digunakan oleh pengguna.

2.2 Metode Analisis

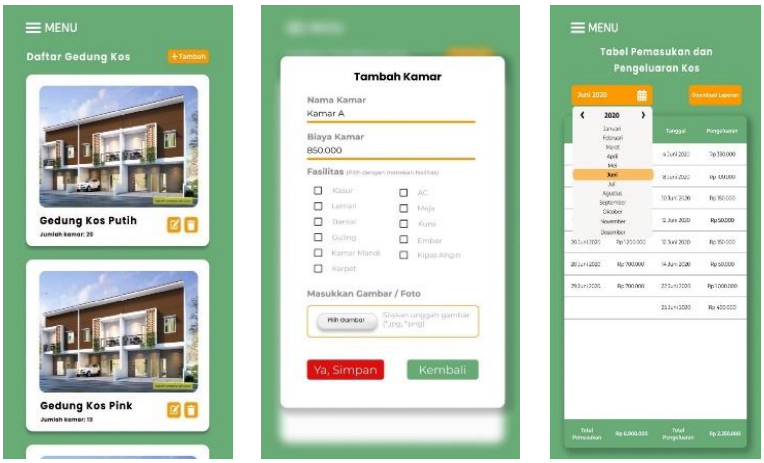
Penulis menggunakan metode observasi sebagai metode untuk mengetahui bagaimana penerapan integrasi antara model Waterfall dan UCD dijalankan oleh programmer. Hal yang ingin diketahui adalah sebagai berikut: (a) Apakah seluruh proses UCD dijalankan pada fase Identifikasi Kebutuhan di model Waterfall? (b) Apakah keseluruhan proses baik proses di dalam model Waterfall dan UCD dijalankan dan sesuai rencana? (c) Apakah ada proses yang memerlukan iterasi? (d) Apakah produk sesuai dengan harapan pengguna? (e) Apa kendala yang dihadapi saat pengembangan perangkat lunak? Observasi dilakukan terhadap dua orang pengembang aplikasi yang berperan sebagai sistem analis, desainer antarmuka, programmer, dan tester aplikasi. Observasi dilakukan dari bulan Februari hingga Juni 2024. Pengamatan dilakukan dengan cara mencatat hal yang penting dan objektif. Setelah observasi selesai, data dikumpulkan untuk dianalisis untuk menjawab kelima pertanyaan tersebut.

3. Hasil dan Pembahasan

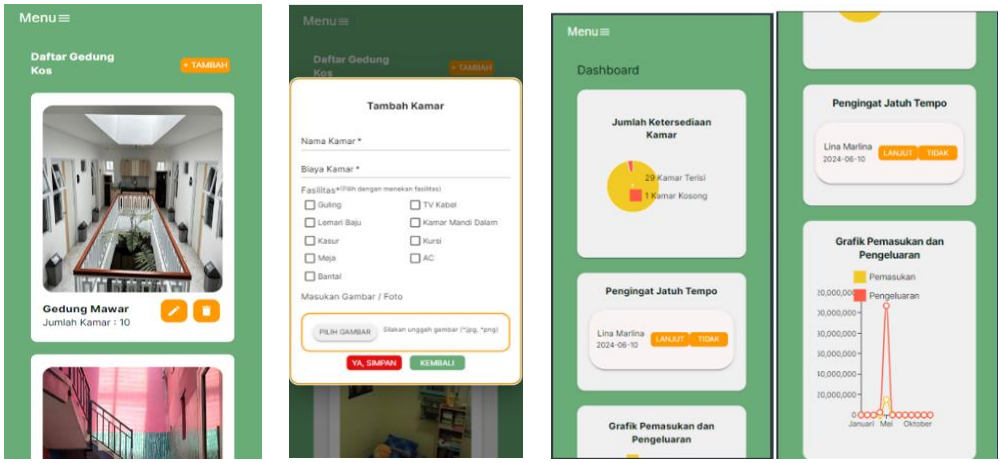
3.1 Produk yang Dihasilkan

Aplikasi pengelolaan indekos berhasil dibangun dan dideploy meski pun tidak terimplementasi secara keseluruhan. Tujuan utama dari aplikasi ini adalah untuk meningkatkan efisiensi dan akurasi dalam operasi sehari-hari indekos, sehingga pemilik indekos dapat dengan mudah mengelola gedung dan fasilitas, pemasukan dan pengeluaran, penyewa, serta jatuh tempo sewa dengan lebih efisien. Teknologi yang digunakan untuk membangun aplikasi ini adalah Laravel dan React. Persentase fitur yang berhasil diimplementasi adalah sebesar 87% dengan detail sebagai berikut: (a) Fitur autentikasi terimplementasi 90%. Fitur ini digunakan untuk pengguna agar bisa masuk ke sistem. Fitur ini terdiri dari fitur login, logout, lupa password, dan registrasi. (b) Fitur pengelolaan gedung dan fasilitas terimplementasi 60%. Fitur Implementasi mencakup penambahan, pembaruan, dan penghapusan data gedung dan fasilitas indekos oleh pemilik. Fasilitas hanya dapat ditambahkan jika gedung sudah terdaftar dalam sistem. Pemilik dapat menambahkan fasilitas yang telah dikategorikan oleh sistem, yaitu kamar, fasilitas umum, dan fasilitas kamar. (c) Fitur Tampilan Laporan Pemasukan dan Pengeluaran terimplementasi 100%. Implementasi mencakup visualisasi tabel pemasukan dan pengeluaran dengan bulan dan tahun yang dapat dipilih sesuai dengan kebutuhan pengguna. Selain itu, dapat mengunduh pemasukan dan pengeluaran indekos dengan berbagai format file yang telah disediakan sepeti CSV, Excel, dan Pdf. (d) Fitur Kelola Penyewa terimplementasi 100%. Implementasi mencakup penambahan, pembaruan, dan penghapusan data penyewa yang dilakukan oleh pemilik. Pemilik dapat melihat status penyewa yang aktif melakukan sewa atau tidak di dalam sistem. (e) Fitur Tampilan Informasi Indekos terimplementasi 70%. Implementasi mencakup visualisasi tampilan ketersediaan kamar, pemberitahuan penyewa yang telah jatuh tempo, dan grafik pemasukan dan pengeluaran yang terjadi dalam indekos. Selain itu, pemilik dapat mengkonfirmasi ke dalam sistem kepada penyewa yang telah jatuh tempo namum akan memperpanjang masa sewa atau tidak. (f) Fitur Kelola Profil Akun terimplementasi 100%. Implementasi mencakup pembaruan dan penghapusan akun pemilik yang terdaftar dalam sistem.

Pada Gambar 3 merupakan beberapa tangkapan layar untuk desain antarmuka dan Gambar 4 merupakan beberapa tangkapan layar untuk antarmuka yang telah terimplementasi. Dapat dilihat bahwa programmer dapat mengimplementasikan desain antarmuka dengan cukup baik dikarenakan hasil implementasi mirip dengan desain antarmuka yang dibuat.



Gambar 3. Tangkapan layar desain antarmuka

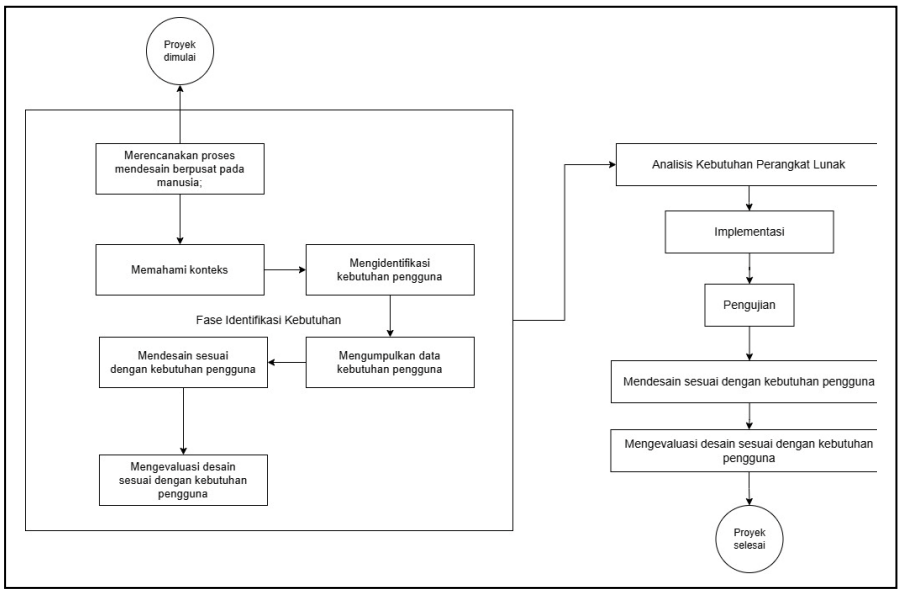


Gambar 4. Tangkapan layar antarmuka yang sudah diimplementasikan

3.2 Proses Pengembangan Perangkat Lunak

Pelaksanaan proses pengembangan dilaksanakan seluruhnya. Dari mulai identifikasi kebutuhan dan penerapan UCD, analisis, desain, implementasi, pengujian dan *deployment*. Langkah-langkah proses implementasi dapat dilihat pada Gambar 5. Proyek dimulai dan pengembang aplikasi merencanakan proses mendesain berpusat pada manusia. Pada tahap ini, pengembang membuat rencana untuk membuat instrumen wawancara dan survei. Pengembang mencari tahu mengenai indekos sekitar. Setelah itu, pada tahap mengidentifikasi kebutuhan pengguna, dilaksanakan survey awal sebanyak 15 pengelola kosan dan 2 kali wawancara terhadap 5 pengelola kosan. Wawancara kedua dilakukan untuk menggali lebih dalam permasalahan pengguna. Dari hasil wawancara dan survey tersebut yang merupakan tahap mengumpulkan data kebutuhan pengguna, dilakukan *user analysis* yang menghasilkan *user persona* dan kebutuhan pengguna. Selanjutnya dilakukan tahap mendesain sesuai kebutuhan pengguna dengan cara mendesain *low fidelity* yang berupa *wireframe* hingga prototipe yang bersifat *high fidelity*.

Setelah desain selesai dibuat, tahap selanjutnya adalah mengevaluasi desain apakah sesuai dengan kebutuhan pengguna dengan cara dievaluasi secara heuristik oleh seorang UI desainer. Rencananya akan dievaluasi oleh dua orang desainer antarmuka, namun, dikarenakan ada permasalahan teknis, sehingga pengembang hanya dapat menghubungi satu orang saja. Evaluasi heuristik yang dilakukan menggunakan pendekatan 10 prinsip Neilsen [17]. Prinsip tersebut diantaranya: (a) *Visibility of System Status*: memastikan bahwa pengguna selalu mengetahui apa yang sedang terjadi dalam sistem; (b) *Match Between System and the Real World*: membuat istilah dan konsep dalam aplikasi sesuai dengan dunia nyata; (c) *User Control and Freedom*: memungkinkan pengguna untuk mudah membatalkan tindakan dan mengontrol aplikasi; (d) *Consistency and Standards*: menjaga konsistensi dalam penggunaan istilah, desain, dan perilaku antarmuka; (e) *Error Prevention*: mencegah terjadinya kesalahan sebelum terjadi; (f) *Recognition Rather Than Recall*: mengurangi beban memori pengguna dengan membuat elemen antarmuka yang dapat dikenali. (g) *Flexibility and Efficiency of Use*: menyediakan fleksibilitas dan efisiensi untuk pengguna pemula dan berpengalaman. (h) *Aesthetic and Minimalist Design*: memastikan desain tetap sederhana dan hanya menampilkan informasi yang relevan. (i) *Help Users Recognize, Diagnose, and Recover from Errors*: membantu pengguna mengenali, mendiagnosis, dan memperbaiki kesalahan. (j) *Help and Documentation*: menyediakan bantuan dan dokumentasi yang mudah diakses. Setiap halaman antarmuka dievaluasi dan dinilai tingkat keparahan (severity level).



Gambar 5. Langkah-langkah proses implementasi

Ada empat tingkat keparahan: *cosmetic problem, minor usability problem, major usability problem, dan usability catastrophe*. Gambar 6 merupakan contoh formulir evaluasi heuristik yang telah diisi oleh desainer antarmuka. Dari enam fitur utama, ada 25 *task* yang dievaluasi secara heuristik. Total terdapat 33 isu heuristik dengan rata-rata 1,32 isu untuk setiap *task*. Isu heuristik yang paling sering muncul di setiap task adalah *Consistency and Standard* yang muncul dalam 16 *task*, disusul oleh *Visibility of System Status* yang muncul pada 8 *task*, dan *Aesthetic and Minimalist Design* yang muncul pada 5 task. Desain antarmuka selanjutnya direvisi sesuai dengan isu-isu yang muncul, namun dikarenakan waktu proyek yang terbatas, tidak dilakukan kembali evaluasi heuristik, namun langsung ke fase analisis kebutuhan perangkat lunak.

Evaluasi fitur: Login dan Register
Evaluasi task: Login dan Register ke aplikasi

No	Judul *Issue*	Deskripsi *Issue*	*Usability Heuristics*	*Severity Level*	Gambar (*Screenshot Issue*)
1	Penambahan fitur "*Forgot Password*" / "Lupa Kata Sandi"	Pada *login and register page* selain ada fitur untuk daftar (*sign in)* dan masuk (*log in)* diperlukan juga fitur untuk menangani pengguna yang memiliki kemungkinan lupa kata sandi / salah kata sandi (*forgot password)* saat ingin *login* ke akunnya.	Error Prevention	4	
2	Posisi logo dan form Login kurang proporsional	Pada *login page,* dapat terlihat bahwa penempatan logo dan form untuk *login* akun masih terlalu ke kanan (sebaiknya terletak pas di tengah)	Consistency and Standards	1	

*(*baris dan kolom dapat disesuaikan)*

Gambar 6. Formulir evaluasi heuristik

Pada fase analisis, dilakukan analisis kebutuhan perangkat lunak yang menghasilkan SRS. Pada tahap implementasi dilakukan pemrograman aplikasi berdasarkan SRS dan desain antarmuka yang telah dibuat. Pada tahap pengujian dilakukan *Unit Testing* dan *User Acceptance Testing* (UAT). Metode yang digunakan untuk melakukan *Unit Testing* adalah Black-box ECP. Pengujian dilakukan kepada 10 orang pemilik indekos. Pengujian dari 91 *test-case* berhasil dieksekusi dan berstatus *pass,* artinya tidak ada fitur yang gagal dijalankan. Begitu pun dengan hasil UAT, yang mengungkapkan bahwa seluruh *task* yang berjumlah 25 task berhasil tanpa ada kegagalan. Namun, dari hasil wawancara terhadap pemilik indekos, pengguna merasa aplikasi kurang nyaman digunakan dari sisi desain antarmuka ditambah dengan implementasi tidak terlaksana 100%. Selanjutnya dalam waktu beberapa hari, dilakukan kembali evaluasi heuristik oleh desainer antarmuka yang berbeda. Lalu, desain direvisi berdasarkan hasil evaluasi tersebut. Implementasi pemrograman berdasarkan desain yang baru tidak dapat diimplementasikan karena waktu pengerjaan proyek telah selesai. Begitu pun dengan *deployment* belum sempat dilaksanakan.

3.3 Integrasi antara Model Waterfall dan UCD

Berikut adalah hasil wawancara terhadap dua orang programmer yang mengimplementasikan aplikasi pengelolaan indekos. Wawancara berdasarkan pertanyaan-pertanyaan yang berhasil diidentifikasi pada bagian 2.2 metode analisis.

Apakah seluruh proses UCD dijalankan pada fase Identifikasi Kebutuhan di model Waterfall?

Proses UCD yang berada pada fase Identifikasi Kebutuhan tidak sepenuhnya terlaksana. Dari 7 tahapan (bisa dilihat pada sub bab 2.1), tahapan ke-6 yakni evaluasi desain tidak terlaksana dengan baik. Evaluasi heuristik telah dilaksanakan oleh seorang desainer antarmuka namun, umpan balik pengguna terhadap desain antarmuka tidak dilaksanakan. Selain itu, hasil evaluasi heuristik yang dilaksanakan pada tahap ini kurang mampu mengevaluasi antarmuka yang dapat memuaskan pengguna. Hal ini berakibat fatal di mana, pada saat UAT, pengguna menilai antarmuka aplikasi kurang nyaman untuk digunakan. Iterasi proses UCD tidak terlaksana pada fase Identifikasi Kebutuhan karena tidak ada umpan balik dari pengguna. Malah, iterasi proses UCD terlaksana setelah UAT dilaksanakan di mana UAT merupakan fase pengujian di model Waterfall. Pada Gambar 5 terlihat jelas proses UCD mendesain dan mengevaluasi desain dilakukan lagi setelah fase pengujian dan proyek selesai tanpa ada perubahan implementasi.

Apakah keseluruhan proses baik proses di dalam model Waterfall dan UCD dijalankan dan sesuai rencana?

Proses implementasi UCD dan Waterfall tidak berjalan sesuai rencana. Desain antarmuka kembali diganti setelah fase pengujian. Desain antarmuka tidak sempat diimplementasikan ke pemrograman, pengujian, dan *deployment* dikarenakan periode pengerjaan terbatas sampai akhir Juni 2024.

Apakah ada proses yang memerlukan iterasi?

Jika melihat runutan proses, iterasi diperlukan untuk memastikan desain antarmuka benar-benar sesuai dengan kebutuhan pengguna dan mementingkan kenyamanan pengguna. Desain harus dipastikan kembali kepada pengguna untuk menerima umpan balik. Iterasi juga diperlukan ketika pengujian telah dilaksanakan karena prototipe desain dan desain yang telah diimplementasikan menjadi aplikasi dapat terasa berbeda. Namun, dikarenakan model yang digunakan adalah Waterfall, perbaikan desain, implementasi, dan pengujian sulit dilakukan. Terbukti dengan proyek ini di mana waktu dan sumber daya yang terbatas, sehingga aplikasi tidak selesai hingga tahap *deployment* sesuai rencana.

Apakah produk sesuai dengan harapan pengguna?

Berdasarkan hasil UAT, fitur-fitur produk yang telah diimplementasikan dengan baik menjawab harapan pengguna. Namun, dari sisi desain antarmuka kurang nyaman digunakan. Kekurangan terbesar dari desain antarmuka yang dibuat adalah belum familier, estetik, tulisan terlalu kecil, dan kecocokan warna.

Apa kendala yang dihadapi saat pengembangan perangkat lunak?

Desainer antarmuka, sistem analis, pemrograman, penguji adalah dua orang yang sama. Pada saat pengembangan, kendala yang dihadapi adalah terlalu terburu-buru ingin mengimplementasikan desain ke dalam kode pemrograman. Fase identifikasi kebutuhan yang menerapkan UCD ternyata memakan waktu yang terlalu banyak, melebihi perkiraan. Kendala selanjutnya adalah pencarian ahli desainer antarmuka yang sulit dilakukan untuk keperluan evaluasi heuristik. Pencarian ahli desainer antarmuka sebaiknya dilakukan pada tahap perencanaan dalam proses UCD.

3.4 Diskusi

Integrasi model Waterfall dan UCD sulit dilakukan berdasarkan proyek pengembangan aplikasi indekos yang dilakukan pada penelitian ini. Tahap UCD bagian evaluasi desain berdasarkan kebutuhan pengguna merupakan tahap yang sangat krusial. Pada penelitian ini, evaluasi heuristik dilakukan untuk melakukan evaluasi desain namun dikerjakan hanya satu kali dan ketika ada revisi desain, tidak dilakukan evaluasi heuristik kembali, sehingga menyebabkan desain yang dihasilkan kurang mencerminkan kenyamanan dan kebutuhan pengguna. Evaluasi heuristik ini seyogyanya dilakukan lebih dari satu orang agar lebih objektif dan lebih profesional. Ahli yang kurang kompeten membuat proses UCD seperti tidak bermasalah dan masuk ke fase berikutnya pada model Waterfall. Namun, pada fase pengujian, masalah pun kembali mencuat karena pada akhirnya, desain antarmuka dapat dievaluasi secara nyata oleh pengguna saat pengujian UAT. Hasil UAT menyatakan bahwa fitur-fitur telah diimplementasikan meski tidak 100% dan sesuai dengan kebutuhan pengguna, namun dari sisi desain antarmuka kurang nyaman digunakan. Hal ini mengakibatkan antarmuka kembali di desain ulang dan pengimplementasian desain pun kembali dilakukan. Model Waterfall dapat dikatakan gagal diterapkan karena fase kembali lagi dari fase pengujian ke fase identifikasi kebutuhan. Dari penelitian ini, dapat disimpulkan bahwa integrasi UCD dan Waterfall sebaiknya tidak dilakukan. UCD sebaiknya diterapkan pada siklus pengembangan perangkat lunak yang lebih fleksibel dari sisi iterasi seperti metode iteratif atau *agile*.

4. Kesimpulan

Tujuan dari integrasi antara model Waterfall dan pendekatan UCD ialah dapat menghasilkan proses pengembangan perangkat lunak yang lebih responsif terhadap kebutuhan pengguna tanpa mengorbankan struktur tahapan yang sistematis dari model Waterfall. Dengan menggabungkan UCD pada tahap analisis dan desain dalam model Waterfall, tim pengembang dapat lebih efektif mengidentifikasi dan mengakomodasi kebutuhan pengguna sejak awal proses, sehingga menghasilkan produk akhir yang lebih *user-friendly*. Namun, penelitian ini menunjukkan hasil sebagai berikut: (a) integrasi ini sulit dilakukan. Faktor paling utama yang membuat ini sulit dilakukan adalah terjadi revisi desain antarmuka yang banyak yang membuat proses pengembangan kembali ke fase awal, identifikasi kebutuhan. Iterasi pada proses UCD dapat memakan waktu lebih banyak dari perkiraan sehingga berisiko bertambah lamanya proyek pengembangan dilaksanakan. (b) evaluasi desain antarmuka pada proses UCD di fase identifikasi kebutuhan sangat krusial karena menentukan apakah desain antarmuka dapat sesuai dengan kebutuhan pengguna dan mementingkan kenyamanan pengguna. (c) evaluasi desain

antarmuka yang kurang optimal dilaksanakan dapat mengakibatkan aplikasi yang telah diimplementasikan tidak diterima oleh pengguna. Sehingga, proses pengembangan pun tidak berlanjut ke fase *deployment*, namun kembali ke fase identifikasi kebutuhan dan masuk ke proses UCD kembali, ke tahap evaluasi desain antarmuka. Padahal, pada model Waterfall, proses tidak dapat kembali lagi ke fase awal. (d) penelitian ini memberikan gambaran yang jelas bagaimana metode penerapan integrasi antara model Waterfall dan UCD yang dapat dilihat pada sub-bab 2.1.

Penelitian ini memiliki keterbatasan dari sisi pengembang proyek, waktu proyek, dan skala proyek. Penelitian lebih lanjut dapat dilakukan dengan mengimplementasikan metode integrasi yang disebutkan pada penelitian ini dalam proyek yang memiliki sumber daya yang cukup baik. Penelitian lebih lanjut dapat dilakukan untuk menguji efektivitas integrasi ini pada proyek yang lebih kompleks dan dalam konteks yang berbeda, termasuk pengembangan perangkat lunak dengan persyaratan keamanan yang ketat atau lingkungan kerja yang lebih dinamis. Selain itu, dapat dilakukan penerapan integrasi untuk model lainnya seperti integrasi antara model iteratif atau Agile dengan proses UCD, sehingga dapat disimpulkan model yang paling tepat untuk menerapkan UCD.

References

[1] E. Salinas, R. Cueva, and F. Paz, "A Systematic Review of User-Centered Design Techniques," in *International Conference on Human-Computer Interaction*, 2020, pp. 253–267. doi: 10.1007/978-3-030-49713-2_18.

[2] J.-Y. Mao, K. Vredenburg, P. W. Smith, and T. Carey, "The state of user-centered design practice," *Commun ACM*, vol. 48, no. 3, pp. 105–109, Mar. 2005, doi: 10.1145/1047671.1047677.

[3] M. I. Ghozali, A. C. Murti, and S. Muzid, "Analisis Penggunaan Metode User-Centered Design dalam Peningkatan Akseptabilitas SIMPELMAS," *KLIK: Kajian Ilmiah Informatika dan Komputer* , vol. 4, no. 2, pp. 1200–1206, 2023, doi: 10.30865/klik.v4i2.1317.

[4] D. R. Fahriyah, D. Ikasari, and Widiastuti, "Implementasi Re-design Application Mobile MRT Jakarta Menggunakan Metode User Centered Design," *Journal of Applied Computer Science and Technology*, vol. 5, no. 1, pp. 98–108, Jun. 2024, doi: 10.52158/jacost.v5i1.812.

[5] A. Mizani and F. F. Lubis, "Evaluasi dan Perbaikan Desain Interaksi Edunex dengan Pendekatan User-Centered Design," *Jurnal Sistem Cerdas*, vol. 5, no. 2, pp. 83–91, 2022.

[6] J. S. Kurnia and M. Awaludin, "PENERAPAN METODE UCD (USER CENTERED DESIGN) SISTEM INFORMASI PENGGAJIAN KARYAWAN BERBASIS WEB PADA KOPERASI KARYAWAN AIR TIMUR JAKARTA (KOPKAR-ATJ)," *Jurnal Sistem Informasi Universitas Suryadarma*, vol. 10, no. 2, pp. 130–138, 2023.

[7] T. A. Felicia, R. Fauzi, F. Mufied, and F. M. Al Anshary, "Perancangan UI/UX Aplikasi Crowdfunding Syariah Untuk UMKM Menggunakan Metode User-Centered Design," *KLIK: Kajian Ilmiah Informatika dan Komputer* , vol. 4, no. 1, pp. 42–52, 2023, doi: 10.30865/klik.v4i1.1084.

[8] M. I. Nuriyana and E. S. Budi, "Perancangan User Interface dan User Experience Aplikasi Pemandu Wisata Kebun Binatang Menggunakan Metode User Centered Design," *RESOLUSI : Rekayasa Teknik Informatika dan Informasi* , vol. 4, no. 1, pp. 75–84, 2023, [Online]. Available: https://djournals.com/resolusi

[9] B. Sedlmayr, J. Schöffler, H.-U. Prokosch, and M. Sedlmayr, "User-centered design of a mobile medication management," *Inform Health Soc Care*, vol. 44, no. 2, pp. 152–163, Apr. 2019, doi: 10.1080/17538157.2018.1437042.

[10] M. H. Puspita, R. N. Kholiza, and Kurniawan, "Perancangan UI/UX Penerimaan Peserta Didik Baru Berbasis Website Menggunakan Metode User Centered Design," *Media Online*, vol. 4, no. 4, pp. 2342–2356, 2024, doi: 10.30865/klik.v4i4.1722.

[11] M. F. A. Nasrullah, R. Yulius, M. S. Nizan, S. Handayani, and B. Failasuf, "E-Catalog Promotion of Fishermen Business Group with User Centered Design Method," *JISA (Jurnal Informatika dan Sains)* , vol. 7, no. 1, pp. 73–78, 2024.

[12] I. Lestari, V. Andria Kusuma, and M. I. Alfani Putera, "Automasi Surat Perintah Perjalanan Dinas Berbasis Website Menggunakan Metode User Centered Design," *Jurnal Sistim Informasi dan Teknologi*, pp. 88–94, Jan. 2024, doi: 10.60083/jsisfotek.v5i4.327.

[13] N. Ramadlan, S. Wulandari, H. P. Sejati, and A. Suhendar, "Penerapan Metode UCD (User Centered Design) Pada Sistem Perpustakaan Sekolah Berbasis Android," *KLIK: Kajian Ilmiah Informatika dan Komputer* , vol. 4, no. 5, pp. 2430–2441, 2024, doi: 10.30865/klik.v4i5.1803.

[14] R. T. Amanda and R. A. Putri, "Penerapan Metode User Centered Design dalam Sistem Penjualan E-Commerce Application of User-Centered Design Method in E-Commerce Sales System," *Sistemasi: Jurnal Sistem Informasi*, vol. 13, no. 3, pp. 2540–9719, 2024, [Online]. Available: http://sistemasi.ftik.unisi.ac.id

[15] S. Dhandapani, "Integration of User Centered Design and Software Development Process," in *2016 IEEE 7th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, IEEE, Oct. 2016, pp. 1–5. doi: 10.1109/IEMCON.2016.7746075.

[16] ISO 9241-210:2010, "Ergonomics of human-system interaction — Part 210: Human-centred design for interactive systems," *ISO*, vol. 2010, 2010.

[17] A. Lodhi, "Usability Heuristics as an assessment parameter: For performing Usability Testing," in *2010 2nd International Conference on Software Technology and Engineering*, IEEE, Oct. 2010. doi: 10.1109/ICSTE.2010.5608809.