

Pengembangan Aplikasi *Mobile* Manajemen Indekos berbasis Android dengan Metode *Human-Centered Design* (Studi Kasus: Kos Bu Parjo)

Adji Pangestu Wicaksono¹, Agi Putra Kharisma², Riswan Septriayadi Sianturi³

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya
Email: ¹adjipangestu00@student.ub.ac.id, ²agi@ub.ac.id, ³rsianturi@ub.ac.id

Abstrak

Banyak pegawai kantor yang tinggal di kelurahan Cikoko, Pancoran, tinggal di kamar indekos. Kos Bu Parjo adalah salah satu yang menyediakan kamar indekos. Kos Bu Parjo masih melakukan pembukuan tanpa rekam cadang digital. Hasil pembukuan tidak dapat diakses ketika buku catatan tidak berada dalam jangkauan fisik, hilang, terkena air, atau jika terjadi bencana alam. Untuk memecahkan masalah tersebut, dikembangkan aplikasi Android menggunakan *framework* Flutter dan Firebase dengan metode *Human-Centered Design* (HCD). Fitur utama pada aplikasi tersebut adalah manajemen penghuni, catatan keuangan, manajemen gedung, manajemen kamar, manajemen akun pengelola, dan *backup* dan *restore*. Aplikasi mengalami pengujian validasi dengan *black box*, pengujian *usability* dengan *system usability scale*, pengujian *compatibility*, dan pengujian *availability*. Pengujian validasi menyatakan aplikasi berhasil memenuhi semua 51 kebutuhan fungsional yang ditentukan, pengujian *usability* dari 5 subjek penelitian menyatakan aplikasi bersifat *acceptable* dengan nilai keseluruhan 77,75, pengujian *compatibility* menyatakan aplikasi berhasil memenuhi syarat versi Android yang digunakan, dan pengujian *availability* menyatakan fitur pencatatan pada aplikasi berhasil digunakan dalam kondisi *offline*.

Kata kunci: *Human-Centered Design, Flutter, Android, Firebase, manajemen indekos, System Usability Scale*

Abstract

Many office employees who live in Cikoko sub-district, Pancoran, stay in boarding rooms. Kos Bu Parjo is one of those that provides boarding rooms. Kos Bu Parjo still does the bookkeeping without a digital backup. The bookkeeping cannot be accessed when the notebook is not within reach, missing, exposed to water, or if a natural disaster occurs. To solve that problem, an Android application using Flutter framework and Firebase with Human-Centered Design method is developed. The main features of the app are occupant management, financial records, building management, room management, business management's account management, and backup and restore. The application went through validation testing with black box, usability testing with system usability scale, compatibility testing, and availability testing. Validation testing reveals that the application successfully fulfilled all 51 specified functional requirements, usability testing from 5 test subjects reveals the application is acceptable with an overall score of 77.75, compatibility testing reveals that the application successfully fulfilled the used Android version requirement, and availability testing reveals that the application's data recording features are operational even while offline.

Keywords: *Human-Centered Design, Flutter, Android, Firebase, boarding house management, System Usability Scale*

1. PENDAHULUAN

Menemui pendatang dari provinsi lain di kelurahan Cikoko, kecamatan Pancoran, kota Jakarta Selatan sebenarnya relatif mudah. Para pendatang tersebut umumnya datang untuk bekerja sebagai pegawai kantoran dari

perusahaan-perusahaan yang ada di sekitar Pancoran. Pegawai-pegawai tersebut juga umumnya mencari tempat tinggal berupa kamar indekos (umumnya disingkat sebagai "kos").

Salah satu tempat yang menyediakan tempat tinggal jenis ini adalah Kos Bu Parjo. Kos Bu Parjo adalah sebuah usaha indekos yang memiliki sekitar 60 kamar yang dibagi di tiga

bangunan terpisah. Sebagai sebuah usaha, tentunya Kos Bu Parjo memerlukan pembukuan karena adanya pencatatan data usaha indekos dan terjadinya transaksi keluar masuknya uang. Menurut observasi yang dilakukan penulis, pembukuan yang dilakukan oleh Kos Bu Parjo masih berbentuk buku fisik tanpa rekam cadang (backup) digital. Hasil pembukuan tidak dapat diakses sama sekali ketika buku catatan usaha tidak berada dalam jangkauan fisik, hilang, terkena air, atau jika terjadi bencana alam.

Kos Bu Parjo sudah pernah mencoba melakukan pencatatan berbasis komputer. Namun, bagian keuangan usaha tidak terbiasa dengan komputer sehingga pencadangan tidak berjalan dengan baik dan rencana ini diabaikan. Pada kali ini, penulis ingin mencoba menerapkan pencatatan dan pencadangan melalui aplikasi berbasis Android dengan alasan bahwa bagian keuangan usaha sudah sangat terbiasa dengan telepon genggam berbasis Android. Selain itu, penerapan menggunakan *Human-Centered Design* (HCD). Dengan digunakannya HCD, diharapkan bagian keuangan usaha ikut terlibat dengan pengembangan aplikasi secara langsung dan lebih mengenali aplikasi yang dihasilkan sehingga aplikasi dapat digunakan tanpa masalah.

Penulis melakukan penelitian awal berupa pemasangan dan pengujian singkat beberapa aplikasi yang tersedia di Play Store. Dari aplikasi-aplikasi yang ada, penulis menemukan beberapa kekurangan. Kekurangan-kekurangan tersebut di antaranya adalah:

1. Memerlukan pendaftaran akun (komitmen yang cukup tinggi karena penghapusan akun beserta datanya belum tentu mungkin dilakukan jika ingin berpindah platform),
2. Aplikasi memiliki iklan yang cukup mengganggu,
3. Aplikasi kurang fleksibel (pada salah satu aplikasi terdapat proses memasukkan tanggal masuk penghuni kos, yang diisi berdasarkan waktu entri dibuat, yang tidak bisa diganti sama sekali, sehingga tidak bisa memasukkan data penghuni lama).

Penulis berniat untuk membuat aplikasi manajemen indekos untuk Kos Bu Parjo yang berbasis Android sebagai alat pencatatan dan rekam cadang digital yang dapat dioperasikan oleh bagian keuangan usaha. Dampak lain yang diharapkan dapat dihasilkan dari digunakannya aplikasi ini adalah pendataan usaha indekos menjadi lebih mudah dan nyaman jika

dibandingkan dengan pencatatan tradisional yang masih menggunakan buku.

Dalam sebuah sistem serupa yang sudah pernah dibuat (Syam, 2018), di dalam aplikasi manajemen kos terdapat dua jenis hak akses. Yang pertama adalah administrator sebagai pemilik atau orang yang bertugas untuk mengelola, memeriksa, dan memasukkan data-data. Yang kedua adalah *user* atau *member*, pegawai atau orang yang dapat mengakses data dari sistem manajemen indekos tanpa hak akses lain.

Penulis berencana menggunakan Flutter sebagai *development kit* yang akan digunakan. Flutter dapat menghasilkan UI yang rapi dengan cepat sehingga pengembangan dapat difokuskan untuk menghasilkan fitur-fitur yang diinginkan. Selain itu, Flutter juga bersifat *multi-platform* sehingga pengembangan aplikasi ini ke depannya dapat diperluas ke perangkat lain seperti iOS dan *desktop*. Penulis berencana menggunakan Firebase sebagai penyedia penyimpanan rekam cadang untuk menghindari kehilangan data di perangkat pengguna.

2. LANDASAN KEPUSTAKAAN

2.1 Indekos

Indekos adalah sebuah rumah, kamar, atau bangunan yang disewakan kepada pihak lain sebagai tempat tinggal/pemondokan dan mengenakan pembayaran sebagai imbalan dalam jumlah tertentu (Wandayu & Pusposari, 2015). Bangunan indekos tidak jarang memiliki seorang penjaga kos sebagai orang yang bertugas memelihara bangunan agar tetap bersih dan mengabari pemilik jika terjadi sesuatu (seperti adanya orang yang ingin menjadi penghuni, atau adanya kerusakan kamar).

2.2 Human-Centered Design

Human-Centered Design (HCD) adalah suatu desain yang bertujuan untuk melibatkan pengguna selama proses perancangan desain (Campese, Amaral, & Mascarenhas, 2020). Keterlibatan pengguna dapat terjadi di beberapa pendekatan berbeda, seperti *design for user*, *design with user*, *design by user*, dan *adaptation by individual user*. Selain itu, juga terdapat tiga strategi untuk mendapatkan keterlibatan pengguna dalam pengembangan suatu produk, yakni *informative* di mana pengguna menyediakan dan/atau menerima informasi, *consultative* di mana pengguna memberikan

komentar, dan *participative* di mana pengguna mempengaruhi keputusan yang berhubungan dengan sistem secara keseluruhan.

Menurut buku “*The Field Guide to Human-Centered Design*” (IDEO.org, 2015), HCD adalah pendekatan yang cukup unik dalam memecahkan masalah, karena terkadang dapat terasa seperti “kegilaan” dibandingkan sebuah metode. Proses pada HCD didesain untuk membuat seseorang belajar langsung dari orang lain dan membuka diri sendiri ke kemungkinan kreatif yang ada. Perancangan dengan HCD dimulai dari observasi hingga pemikiran abstrak (*diverging*) kemudian kembali lagi ke pengerjaan *prototype* yang sedang dibuat (*converging*). Proses ini diulang secara terus menerus sehingga seseorang dapat mengidentifikasi ide apa yang paling tinggi memiliki kemungkinan untuk berhasil.

2.3 System Usability Scale

System Usability Scale (SUS) adalah sebuah skala sederhana dengan 10 butir pertanyaan yang memberikan pandangan secara keseluruhan mengenai penilaian *usability* secara subjektif (Jordan, Thomas, WeerdMeester, & McClelland, 1996). Kumpulan dari 50 butir kuesioner potensial dikumpulkan, dan digunakan untuk dua contoh perangkat lunak (dengan basis persetujuan umum di mana salah satunya dianggap sangat mudah untuk digunakan dan yang lain hampir tidak mungkin untuk digunakan), salah satunya ditujukan untuk pengembang sistem, dan yang lain untuk pengguna umum.

Butir-butir yang mengarah ke respons ekstrem dari kumpulan pernyataan awal kemudian dipilih. Butir-butir yang terpilih memiliki interkorelasi yang sangat dekat. Selain itu, butir-butir dipilih sedemikian rupa agar setengah dari respons umumnya adalah sangat setuju dan setengah lainnya sangat tidak setuju. Hal ini dilakukan untuk mencegah bias respons yang disebabkan oleh responden yang tidak memikirkan tiap pernyataan dengan baik.

Nilai hasil dari SUS dapat dibandingkan dengan penilaian *acceptability range* (Bangor, Kortum, & Miller, 2009). Nilai di bawah 60,0 dapat dikatakan sebagai *not acceptable*, dan nilai di atas 60,0 dapat dikatakan sebagai *acceptable*.

2.4 Android

Android adalah sebuah platform *open source* yang dirancang untuk perangkat *mobile*

(Gargenta, 2011). Android merupakan suatu platform komprehensif, yang berarti Android sendiri memiliki *stack* perangkat lunak yang lengkap untuk perangkat *mobile*. Bagi pengembang, Android menyediakan semua perlengkapan dan *framework* seperti Android SDK yang diperlukan untuk mengembangkan perangkat lunak *mobile* dengan cepat dan mudah.

2.5 Flutter

Flutter adalah seperangkat perkakas yang mengizinkan pengguna untuk membuat aplikasi lintas platform yang dapat dijalankan di iOS, Android, dan Web (Payne, 2019). Flutter sendiri sebenarnya tidak difokuskan khusus untuk pengembangan Android, namun untuk pengembangan *mobile* secara keseluruhan. Sehingga, Flutter dalam konteks ini digunakan untuk memudahkan pengembangan jika di masa depan ingin dilakukan ekspansi ke aplikasi berbasis web atau iOS.

2.6 Firebase

Firebase adalah sebuah kontainer untuk aplikasi lintas Apple, Android, dan platform perangkat lunak web (Google, 2022). Firebase mendukung berbagai fitur seperti autentikasi, *database*, pengaturan, dan notifikasi antar aplikasi *cross-platform* yang digunakan, yang dapat diakses melalui sebuah *control panel* yang bernama Firebase Console.

Salah satu fitur yang disediakan adalah Firebase Storage, yang berfungsi sebagai layanan penyimpanan konten yang diunggah oleh pengguna seperti gambar, video, atau jenis konten lainnya. Layanan yang disediakan oleh Firebase Storage mengizinkan *developer* untuk mengunduh dan mengunggah *file* langsung dari sisi pengguna. Jika kondisi jaringan pengguna kurang memadai, maka pengguna dapat mencoba operasi pengunduhan atau pengunggahan langsung dari titik di mana masalah jaringan dimulai, sehingga menghemat waktu dan *bandwidth*.

Fitur yang disediakan untuk autentikasi pengguna adalah Firebase Auth. Firebase Auth menyediakan layanan *backend*, SDK, dan *library* antarmuka yang dapat digunakan untuk melakukan autentikasi pengguna suatu aplikasi. Dukungan autentikasi di antaranya adalah melalui *password*, nomor telepon, akun Google, akun Facebook, dan akun Twitter.

Manajemen pengguna pada Firebase Auth

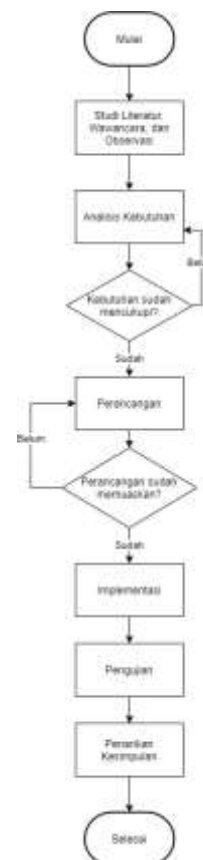
menggunakan SDK khusus bernama Firebase Admin SDK. SDK ini mengizinkan *developer* untuk mengintegrasikan server pribadi dengan Firebase Auth. Dengan menggunakan Firebase Admin SDK, *developer* dapat melakukan manajemen pengguna seperti menambah, mengedit, dan menghapus pengguna yang ada tanpa harus menggunakan *token* autentikasi milik pengguna tersebut. Selain itu, *developer* dapat menambahkan *custom claims* yang berfungsi untuk menambahkan *claim* khusus pada tiap akun. Contoh *claim* yang dimaksud seperti menambahkan tanda khusus pada pengguna yang menjadi seorang administrator.

2.7 Prototyping

Prototyping adalah salah satu pendekatan SDLC terbaik jika pelanggan memiliki tujuan secara umum namun tidak terdapat kebutuhan fungsi dan fitur secara mendetail (Pressman & Maxim, 2020). Paradigma *Prototyping* dimulai pada tahap *communication*. Pengembang akan bertemu dengan *stakeholder* untuk menentukan tujuan akhir, mengidentifikasi kebutuhan yang diketahui, dan memberikan garis besar di tempat yang sekiranya dibutuhkan. Iterasi *prototyping* direncanakan dengan cepat yang akan menghasilkan sebuah model dengan bentuk *quick design*. *Quick design* berfokus pada representasi dari aspek perangkat lunak yang akan terlihat oleh pengguna seperti *layout* dari *user interface*.

3. METODOLOGI PENELITIAN

Penelitian ini menggunakan metode *Human-Centered Design*. Subjek dari penelitian yang dilaksanakan adalah Kos Bu Parjo, sebuah usaha indekos di Cikoko. Perancangan dan implementasi perangkat lunak penelitian menggunakan pendekatan *software development life cycle (SDLC) prototyping*, yang memanfaatkan komunikasi yang cukup sering antara peneliti dengan subjek penelitian akibat penggunaan HCD. Diagram alir penelitian ini dapat dilihat pada Gambar 1.



Gambar 1. Diagram Alir Penelitian

Penelitian ini diawali dengan studi literatur untuk memperdalam pemahaman penulis. Wawancara dan observasi untuk pengumpulan data dilakukan sebelum analisis kebutuhan dilakukan.

HCD diterapkan saat melakukan identifikasi kebutuhan pada tahap analisis kebutuhan. Hasil observasi dan wawancara yang awalnya bersifat abstrak diubah menjadi kebutuhan-kebutuhan yang bersifat konkret. Setelah kebutuhan didapatkan, akan dikonfirmasi ke pihak subjek penelitian. Jika terdapat kekurangan dari kebutuhan yang ada, maka akan ditambahkan kebutuhan baru ke dalam aplikasi.

Pada tahap perancangan, dilakukan perancangan dari kebutuhan yang sudah dihasilkan dari pengumpulan data. Data-data yang didapat kemudian diterjemahkan menjadi rancangan dengan menerapkan metode HCD. Hasil perancangan awal akan ditunjukkan ke pihak subjek penelitian untuk pengumpulan pendapat yang dijadikan evaluasi untuk rancangan iterasi selanjutnya. Selain itu, dibuat *pseudo code* dari kebutuhan yang sudah didapatkan.

Pada tahap implementasi, dilakukan pembuatan aplikasi dari rancangan yang telah

dibuat. Tahap ini menghasilkan sebuah aplikasi Android yang sudah bisa diberikan kepada subjek penelitian untuk pengujian.

Pada tahap pengujian, dilakukan tiga jenis pengujian. Pengujian pertama yang dilakukan adalah pengujian validasi menggunakan *black box*. Pengujian validasi akan menguji semua kebutuhan fungsional yang telah didefinisikan. Kemudian, dilakukan pengujian *usability* menggunakan metrik *System Usability Scale* (SUS) untuk mendapatkan nilai *system usability* yang menentukan seberapa *acceptable* sebuah aplikasi. Pengujian selanjutnya adalah pengujian *compatibility* yang dilakukan untuk menguji kemampuan dijalkannya aplikasi yang telah dibuat di sistem operasi Android yang telah ditentukan pada saat penggalan kebutuhan. Pengujian terakhir adalah pengujian *availability* yang menguji kemampuan aplikasi ketika digunakan secara *offline*.

Setelah pengujian selesai, dilakukan penarikan kesimpulan yang diambil dari hasil pengujian yang didapatkan. Kesimpulan yang dibuat menjawab rumusan masalah yang telah ditentukan. Selain itu, terdapat penulisan saran untuk penelitian selanjutnya berdasarkan hal-hal yang masih bisa diperbaiki.

4. ANALISIS KEBUTUHAN

Pada analisis kebutuhan, dilakukan identifikasi aktor dan identifikasi kebutuhan fungsional dan non-fungsional. Aktor yang berhasil diidentifikasi adalah pemilik usaha, pegawai usaha, dan *guest*. Terdapat 51 kebutuhan fungsional, 9 di antaranya adalah milik sistem, 31 milik pemilik usaha, 10 milik pegawai usaha, dan 1 milik *guest*. Terdapat 2 kebutuhan non-fungsional yang berhasil diidentifikasi yakni kebutuhan non-fungsional *usability* dan *compatibility*. Beberapa kebutuhan fungsional yang dihasilkan dapat dilihat pada Tabel 1.

Tabel 1. Kebutuhan Fungsional

No.	Aktor	Nama Kebutuhan	Deskripsi Kebutuhan
1	Sistem	Mencatat pendapatan milik usaha	Aplikasi harus mampu mencatat pendapatan milik usaha
2	Sistem	Mencatat data penghuni kamar kos	Aplikasi harus mampu mencatat data penghuni kamar kos

			berupa nama, alamat asal, tanggal masuk, foto KTP, no. HP, no. telepon keluarga yang dapat dihubungi, dan foto surat nikah (opsional)
3	Sistem	Membedakan jenis tampilan penghuni kos bagi pemilik dan pegawai usaha	Aplikasi harus mampu membedakan data dasar dan data lengkap penghuni kamar kos. Data dasar penghuni kamar kos adalah nama, no. HP, tanggal masuk, dan nomor kamar kos yang ditinggali. Data lengkap penghuni kos adalah data yang didefinisikan pada nomor 2 pada tabel ini
4	Pemilik Usaha	Mengedit pendapatan dan pengeluaran kos	Aplikasi harus mampu menyediakan fitur untuk mengedit pendapatan dan pengeluaran yang sudah ada yang belum dikunci
5	Pemilik Usaha	Melakukan rekapitulasi pendapatan dan pengeluaran kos	Aplikasi harus mampu menyediakan fitur untuk melakukan rekapitulasi untuk pendapatan dan pengeluaran pada tiap akhir bulan. Jika hasil rekapitulasi sudah disetujui, maka data pada bulan tersebut

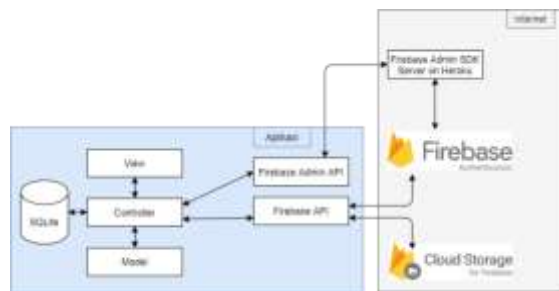
			akan dikunci dan tidak dapat diubah lagi
6	Pegawai Usaha	Melihat daftar penghuni kos	Aplikasi harus mampu menyediakan fitur untuk melihat daftar penghuni kos yang ada di dalam aplikasi
7	Pegawai Usaha	Melihat daftar kamar kos	Aplikasi harus mampu menyediakan fitur untuk melihat daftar kamar kos yang ada di dalam aplikasi
8	Guest	Melakukan login	Aplikasi harus mampu menyediakan fitur login untuk mengubah seorang guest menjadi aktor pemilik usaha atau pegawai usaha

Setelah itu, dilakukan pembuatan *use-case diagram* berdasarkan kebutuhan fungsional dan aktor yang telah ditentukan.

5. PERANCANGAN DAN IMPLEMENTASI

5.1 Perancangan

Perancangan arsitektur sistem yang dibuat dapat dilihat pada Gambar 2. Aplikasi yang dibuat menggunakan arsitektur *model, view, dan controller*. *Model* dibuat sebagai “perwakilan” dari basis data SQLite yang digunakan aplikasi. Pada aplikasi ini, *controller* bertanggung jawab atas akses data antara logika program dan basis data. *Controller* juga akan bertanggung jawab menjadi perantara antar *model* dan *view*. *View* pada aplikasi ini dianggap sebagai bagian yang dapat berinteraksi dengan pengguna.

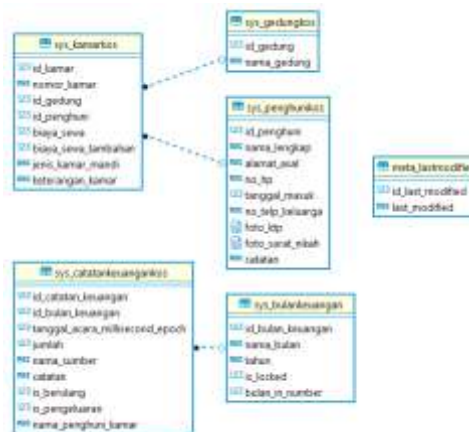


Gambar 2. Arsitektur Sistem

Pada perancangan arsitektur, Firebase API dipisah dengan *controller* untuk memperjelas bahwa yang bertanggung jawab untuk berkomunikasi dengan layanan dari Firebase bukan *controller* secara keseluruhan, namun dengan bantuan Firebase API. Aplikasi ini menggunakan Firebase Authentication untuk melakukan autentikasi pengguna dan Firebase Cloud Storage sebagai layanan untuk pencadangan basis data SQLite yang digunakan. Selain itu, Firebase Admin API dibuat terpisah karena Firebase tidak mengizinkan akses admin API secara lokal, sehingga harus dibuat sebuah *secure environment* di mana di dalamnya dapat dijalankan sebuah server Firebase Admin SDK. Aplikasi ini menggunakan SQLite dengan alasan mempermudah manipulasi data ketika perangkat tidak terhubung dengan internet.

Salah satu jenis perancangan lain yang dilakukan adalah perancangan antarmuka. Perancangan ini menghasilkan desain antarmuka *high-fidelity* yang akan dikonsultasikan dengan pihak subjek penelitian untuk evaluasi.

Perancangan lain yang dilakukan adalah perancangan basis data. Perancangan ini merencanakan skema dari tabel yang digunakan untuk menyimpan data pada aplikasi yang dibuat. Ilustrasi *entity relationship diagram* dari perancangan basis data dapat dilihat pada Gambar 3.



Gambar 3 Entity relationship diagram

5.2 Implementasi

Aplikasi dibuat menggunakan *framework* Flutter dengan bahasa pemrograman Dart. Basis data yang digunakan adalah SQLite. Pada bagian ini, implementasi dipisah menjadi implementasi basis data, implementasi kode program, dan implementasi antarmuka.

Implementasi antarmuka dari halaman daftar pendapatan dan pengeluaran kos dapat dilihat pada Gambar 4.



Gambar 4. Implementasi Antarmuka Halaman Daftar Pendapatan dan Pengeluaran Kos

6. PENGUJIAN DAN PEMBAHASAN

6.1 Pengujian Validasi

Pengujian validasi menguji aplikasi terhadap 51 kebutuhan fungsional yang telah didapatkan menggunakan pengujian *black box*. Kasus uji pengujian validasi untuk 9 kebutuhan fungsional milik sistem beserta hasilnya dapat dilihat pada Tabel 2.

Tabel 2. Hasil Pengujian Validasi

N o.	Kasus Uji	Hasil yang Diharapkan	Hasil yang Didapatkan	Status Validasi
1	Mencatat pendapatan milik usaha	Sistem menampilkan catatan pendapatan usaha yang disimpan.	Sistem menampilkan catatan pendapatan yang disimpan.	Valid
2	Mencatat data penghuni kamar kos	Sistem menampilkan data penghuni	Sistem menampilkan data penghuni	Valid

		kos yang telah dibuat.	kos yang telah dibuat.	
3	Membedakan jenis tampilan penghuni kos bagi pemilik dan pegawai usaha	Sistem menampilkan tampilan yang berbeda untuk akun pemilik usaha dan akun pegawai usaha.	Sistem menampilkan tampilan yang berbeda untuk akun pemilik usaha dan akun pegawai usaha.	Valid
4	Mengedit pendapatan dan pengeluaran kos	Data milik entri yang ditampilkan oleh sistem setelah di-edit berbeda dengan data sebelum di-edit.	Data milik entri yang ditampilkan oleh sistem setelah di-edit berbeda dengan data sebelum di-edit.	Valid
5	Melakukan rekapitulasi pendapatan dan pengeluaran kos	Sistem menampilkan rekapitulasi pendapatan dan pengeluaran yang ada pada bulan yang dipilih. Setelah tombol kunci data dipilih, tombol hapus pada setiap entri data pada bulan yang dipilih tidak ditampilkan.	Rekapitulasi berhasil ditampilkan tanpa masalah. Data yang ada pada bulan yang sudah dikunci datanya tidak menampilkan tombol hapus.	Valid
6	Melihat daftar penghuni kos	Sistem berhasil menampilkan daftar penghuni kos yang ada untuk pegawai usaha.	Sistem berhasil menampilkan daftar penghuni kos yang ada untuk pegawai usaha.	Valid
7	Melihat daftar kamar kos	Sistem berhasil menampilkan daftar kamar kos yang ada	Sistem berhasil menampilkan daftar kamar kos yang ada	Valid

		untuk pegawai usaha.	untuk pegawai usaha.	
8	Melakukan login	Sistem berhasil melakukan login dan restore sehingga pengguna dapat masuk ke dalam aplikasi dengan basis data yang sesuai dengan backup terbaru.	Sistem berhasil melakukan login dan restore, dan data yang digunakan adalah data terbaru.	Valid

Dari pengujian yang dilakukan, semua hasil pengujian menyatakan hasil valid sehingga dapat dinyatakan bahwa sistem yang dibuat telah memenuhi semua kebutuhan fungsional yang ada.

6.2 Pengujian Usability

Pada pengujian *usability* dilakukan analisis menggunakan SUS dengan membagikan satu lembar kuesioner SUS kepada 4 orang pemilik usaha dan 1 orang pegawai usaha. Hasil kuesioner SUS untuk keseluruhan subjek penelitian dapat dilihat pada Tabel 3, dengan pemilik usaha disingkat sebagai PU dan pegawai usaha disingkat sebagai PG.

Tabel 3. Hasil Kuesioner SUS

Penguji	Jumlah Nilai	Jumlah Perkalian (Nilai x 2,5)	Nilai SUS (Jumlah Perkalian / Jumlah Penguji)
PU	127	317,5	79,375
PG	28	70,0	70,0
PU+PG	155	387,5	77,5




Nilai SUS menurut pihak pemilik usaha, pegawai usaha, dan secara keseluruhan secara berturut-turut adalah 79,375; 70,0; dan 77,5. Berdasarkan hasil kuesioner yang didapat, sistem dapat dinyatakan *acceptable* karena memiliki nilai SUS di atas 60,0.

6.3 Pengujian Compatibility

Pengujian *compatibility* dibuat berdasarkan kebutuhan non-fungsional yang telah dibuat yakni “aplikasi harus dapat digunakan pada ponsel genggam dengan sistem operasi minimal Android 8.1 (API level 27) dengan batas akhir

hingga Android 11 (API level 30)”. Sampel dari hasil pengujian ini dapat dilihat pada Tabel 4.

Tabel 4. Hasil Pengujian *Compatibility*

No.	Spesifikasi Perangkat	Screenshot	Status Validasi
1	Xiaomi Redmi Note 7 Versi Android 11 (API level 30)		Valid
2	Samsung Galaxy J7 Versi Android 8.1 (API level 27)		Tidak valid
3	Android Emulator Versi Android 8.1 (API level 27)		Valid

Dari hasil pengujian, dapat dilihat bahwa terdapat satu kasus pengujian yang tidak valid. Kasus tersebut adalah pemasangan aplikasi di ponsel Android milik salah satu subjek penelitian dengan Android versi 8.1 (API level 27). Meskipun demikian, setelah dilakukan pengujian pada Android emulator dengan versi sama (Android 8.1, API level 27), dapat dilihat bahwa aplikasi dapat dipasang tanpa kendala. Maka, dapat diasumsikan bahwa kegagalan yang terjadi merupakan faktor eksternal dan bukan kesalahan dari sistem yang telah dibuat.

6.4 Pengujian Availability

Pengujian *availability* dilakukan untuk menguji kemampuan aplikasi untuk digunakan dalam kondisi *offline*, sesuai dengan kebutuhan non-fungsional *availability*, yakni “fitur-fitur pencatatan pada aplikasi harus dapat digunakan oleh pengguna tanpa koneksi internet”. Beberapa fitur pencatatan diuji pengoperasiannya tanpa koneksi internet dengan membandingkan hasil yang diharapkan dengan skenario alternatif jika terjadi kegagalan.

Tabel 5 Hasil Pengujian Availability

No.	Kasus Uji	Hasil yang Diharapkan	Skenario Alternatif	Hasil yang Didapatkan	Status Validasi
1	Menambahkan bahkan penghunian kos	Sistem berhasil menyimpan data penghunian yang diinputkan sebelumnya meskipun dalam kondisi <i>offline</i> .	Sistem gagal menyimpan data penghunian yang diinputkan sebelumnya ketika sedang dalam kondisi <i>offline</i> .	Sistem berhasil menyimpan data penghunian meskipun dalam kondisi <i>offline</i> .	Valid
2	Melihat penghunian kos (lengkap)	Sistem berhasil menampilkan data penghunian yang dipilih meskipun dalam kondisi <i>offline</i> .	Sistem gagal menampilkan data penghunian yang dipilih ketika sedang dalam kondisi <i>offline</i> .	Sistem berhasil menampilkan data penghunian yang dipilih dalam kondisi <i>offline</i> .	Valid
3	Mengedit penghunian kos	Sistem dapat menyimpan dan menampilkan perubahan data yang dilakukan meskipun dalam	Sistem gagal menampilkan perubahan data jika dibandungkan dengan sebelumnya ketika sedang dalam	Sistem berhasil menyimpan dan menampilkan perubahan data yang dilakukan dalam kondisi <i>offline</i> .	Valid

		kondisi <i>offline</i> .	kondisi <i>offline</i> .		
4	Menghapus penghunian kos	Sistem menghapus entri data penghunian yang dipilih oleh pengguna untuk dihapus meskipun dalam kondisi <i>offline</i> .	Sistem gagal menghapus entri data penghunian yang dipilih oleh pengguna untuk dihapus ketika sedang dalam kondisi <i>offline</i> .	Sistem berhasil menghapus entri data penghunian yang dipilih oleh pengguna untuk dihapus dalam kondisi <i>offline</i> .	Valid

Pada hasil pengujian dapat dilihat bahwa semua hasil yang didapat menyatakan bahwa sistem tetap berhasil melaksanakan fungsi yang dimaksud meskipun dalam kondisi *offline*. Dari hasil tersebut dapat dinyatakan bahwa semua hasil pengujian *availability* dapat dinyatakan sebagai valid.

7. PENUTUP

7.1 Kesimpulan

Aplikasi manajemen Kos Bu Parjo memiliki 51 kebutuhan fungsional dan 2 kebutuhan non-fungsional. Di antara 51 kebutuhan fungsional tersebut, 9 di antaranya adalah kebutuhan fungsional milik sistem, 31 di antaranya adalah kebutuhan fungsional pemilik usaha, 10 di antaranya adalah kebutuhan fungsional pegawai usaha, dan 1 di antaranya adalah kebutuhan fungsional milik guest. Kebutuhan fungsional yang dibuat mencakup kemampuan untuk mengatur penghunian kos, catatan keuangan kos, gedung kos, kamar kos, akun pihak manajemen kos, dan backup dan restore. Selain itu, kebutuhan non-fungsional usability dibuat agar aplikasi dapat digunakan tanpa mengganggu kenyamanan pihak pemilik usaha saat melakukan pencatatan data. Untuk kebutuhan non-fungsional compatibility, hal tersebut dibuat berdasarkan ponsel Android yang digunakan oleh pihak manajemen Kos Bu Parjo.

Implementasi aplikasi dilakukan menggunakan bahasa pemrograman Dart pada framework Flutter dengan dukungan Firebase.

Implementasi aplikasi dilakukan dengan memperhatikan kebutuhan fungsional dan non-fungsional yang telah diketahui. Setelah implementasi selesai, dilakukan pengujian validasi, usability, dan compatibility untuk aplikasi yang sudah dibuat. Pada pengujian validasi, aplikasi mampu melewati semua pengujian kebutuhan fungsional sebanyak 51 kebutuhan. Pada pengujian usability, aplikasi mendapatkan nilai SUS sebesar 79,375 dari pihak pemilik usaha dan 70,0 dari pihak pegawai usaha. Jika dirata-rata, maka nilai SUS secara keseluruhan adalah 77,75 sehingga aplikasi dapat dinyatakan sebagai acceptable karena nilai SUS berada di atas 60,0. Pada pengujian compatibility, aplikasi mengalami kendala ketika dipasang pada versi Android salah satu ponsel pemilik usaha. Namun, kegagalan tersebut merupakan faktor eksternal karena aplikasi tidak mengalami kendala ketika dipasang pada Android emulator dengan versi Android yang sama, sehingga hasil pengujian tetap dianggap valid. Hasil dari pengujian compatibility menyatakan bahwa aplikasi berhasil melewati pengujian compatibility pada versi Android 8.1 (API level 27) hingga Android 11 (API level 30). Pada pengujian *availability*, semua hasil pengujian dinyatakan valid karena fitur pencatatan pada aplikasi dapat digunakan secara normal dalam kondisi *offline*. Setelah semua tahap pengujian dilakukan, aplikasi dinyatakan mampu melewati semua tahap pengujian dan sudah memenuhi semua kebutuhan fungsional dan non-fungsional yang diperlukan oleh pihak Kos Bu Parjo, sehingga dapat dinyatakan bahwa kebutuhan fungsional dan non-fungsional untuk Kos Bu Parjo dapat direalisasikan dalam bentuk aplikasi berbasis Android.

7.2 Saran

Saran yang dapat digunakan untuk pengembangan lebih lanjut adalah:

1. Mengubah sistem ini agar dapat digunakan secara umum, tidak terbatas ke implementasi di lingkungan Kos Bu Parjo.
2. Ditambahkan fitur untuk mengakses arsip penghuni yang pernah ada di suatu kamar kos.
3. Ditambahkan fitur administrasi tambahan seperti pengiriman pengingat tagihan melalui Whatsapp.
4. Memperluas jangkauan aplikasi agar dapat digunakan di web dan di iOS

dengan cara memanfaatkan dukungan yang didapat dari digunakannya framework Flutter.

8. DAFTAR REFERENSI

- Bangor, A., Kortum, P., & Miller, J. (2009). Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale. *Journal of Usability Studies*, 4(3), 114-123.
- Campese, C., Amaral, D. C., & Mascarenhas, J. (2020). Restating The Meaning of UCD And HCD For A New World Of Design Theories. *Interacting With Computers*, 32(1), 33-51.
- Gargenta, M. (2011). *Learning Android* (1st ed.). California: O'Reilly.
- Google. (2022). *Firestore FAQ*. Retrieved 2022, from <https://firebase.google.com/support/faq/>
- IDEO.org. (2015). *The Field Guide To Human-Centered Design* (1st ed.). Canada: IDEO.
- Jordan, P. W., Thomas, B., WeerdMeester, B. A., & McClelland, I. L. (1996). SUS: The System Usability Scale. In *Usability Evaluation in Industry* (pp. 191-194). Taylor & Francis.
- Payne, R. (2019). Hello Flutter. In *Beginning App Development with Flutter* (pp. 3-8). Dallas, Texas, USA: Apress.
- Pressman, R. S., & Maxim, B. R. (2020). *Software Engineering: A Practitioner's Approach* (9th ed.). New York: McGraw-Hill.
- Syam, E. (2018). Rancang Bangun Sistem Informasi Rumah Kost dan Kontrakan Teluk Kuantan. *Jurnal Teknologi dan Open Source*, 1(1).
- Wandayu, H., & Pusposari, D. (2015). Persepsi Pemilik Usaha Kos Terhadap Pemungutan Pajak Penghasilan Pasal 4 Ayat 2 Atas Penghasilan Dari Persewaan Tanah Dan/Bangunan Kategori Rumah Indekos Di Kota Malang. *Jurnal Ilmiah Mahasiswa FEB Universitas Brawijaya*, 4(1).