

Pengembangan Aplikasi Perangkat Bergerak Sistem Manajemen Pembelian dan Patungan Hewan Kurban Berbasis Android (Studi Kasus Masjid Al Huda, Malang)

Reza Agung Nugraha, Mahardeka Tri Ananta, Adam Hendra Brata

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya
Email: rezaagung3910@gmail.com, deka@ub.ac.id, adam@ub.ac.id

Abstrak

Masjid Al Huda, Malang mengalami permasalahan terhadap kurangnya partisipasi masyarakat setempat pada program patungan hewan kurban akibat kurangnya efektivitas manajemen pengelolaan dan metode penyebaran informasi. Hasil survei terhadap 51 umat Islam menunjukkan bahwa meskipun terdapat platform pembelian hewan kurban secara daring, 90,2% responden memilih untuk berkorban dengan membeli atau patungan hewan kurban di masjid terdekat agar dapat melihat fisik hewan kurban secara langsung. Saat ini, aplikasi untuk meningkatkan efektivitas manajemen pembelian dan patungan hewan kurban telah banyak dikembangkan. Namun, belum terdapat aplikasi yang memfasilitasi pembelian dan patungan hewan kurban tanpa mengharuskan jemaah untuk mencari partner patungan secara mandiri. Dengan demikian, dilakukan pengembangan aplikasi yang dapat memajemen dan menjadi media informasi terbaru dari program patungan hewan kurban, serta memungkinkan jemaah untuk tidak mencari partner patungan secara mandiri dan dapat melihat hewan kurban secara langsung melalui fitur pencarian lokasi masjid penyedia hewan kurban terdekat. SDLC *Waterfall* dipilih untuk menjamin pengembangan aplikasi telah terencana dengan baik, sehingga menghasilkan produk aplikasi berkualitas tinggi. Pengujian unit, pengujian validasi, dan pengujian usability telah memastikan sistem memiliki seluruh program dan fungsionalitas yang valid, dapat menyelesaikan masalah secara efektif dan efisien, serta memuaskan pengguna dalam mencapai tujuannya.

Kata kunci: *pembelian, patungan, hewan kurban, android, waterfall, SUS*

Abstract

The Al Huda Mosque in Malang faces challenges due to a lack of local community participation in the joint venture program for qurbani animals, resulting from ineffective management and information dissemination methods. Survey results of 51 Muslims showed that despite the existence of an online qurbani animal purchasing platform, 90.2% of respondents chose to perform qurbani by purchasing or joint ventures qurbani animals at the nearest mosque in order to see the physical qurbani animals directly. Currently, many applications have been developed to enhance the effectiveness of qurbani animal purchase and joint venture management. However, there is no application that facilitates the purchase and joint venture of qurbani animals without requiring worshippers to find joint venture partners independently. As a result, an application was developed that can manage and become the latest information media of the qurbani animal joint venture program, as well as allow worshippers not to look for joint venture partners independently and can see the qurbani animals directly through the location search feature of the nearest mosque providing qurbani animals. The Waterfall SDLC was chosen to ensure that application development is well planned, resulting in high-quality application products. Unit testing, validation testing, and usability testing have ensured the system has all valid programs and functionality, can solve problems effectively and efficiently, as well as satisfy users in achieving their goals.

Keywords: *purchase, joint venture, qurbani animals, android, waterfall, SUS*

1. PENDAHULUAN

Qurban atau dalam bahasa Indonesia adalah “kurban” berarti sebuah ritual persembahan umat Islam kepada Allah sebagai wujud ketaatan melalui penyembelihan hewan ternak pada hari raya Iduladha. Berdasarkan penelitian yang dilakukan oleh Wardi dan Waldi (2021), didapatkan fakta bahwa terdapat cara lain yang halal dalam pembelian hewan kurban, yaitu melalui patungan terhadap hewan sapi, kerbau dan unta dengan jumlah patungan maksimal untuk tujuh orang pada satu ekor hewan kurban.

Berdasarkan hasil wawancara dengan panitia kurban masjid Al Huda yang berlokasi di Jl. Kendalsari, Tulusrejo, Kec. Lowokwaru, Kota Malang, diketahui bahwa masjid ini melaksanakan program patungan hewan kurban sapi setiap tahunnya. Dalam pelaksanaannya, panitia menyebarkan informasi program patungan sejak satu bulan sebelum hari raya Iduladha melalui pengumuman ketika pelaksanaan salat Jumat. Selain itu, panitia akan menawarkan program patungan hewan kurban kepada jemaah yang berpartisipasi di program patungan tahun sebelumnya satu per satu, baik melalui aplikasi pesan maupun mendatangi rumah secara langsung.

Metode konvensional yang digunakan oleh masjid Al Huda dalam program patungan hewan kurban menimbulkan permasalahan. Tidak tersedianya media informasi terkini yang disertai dengan kurangnya efektivitas metode penyebaran informasi menyebabkan terbatasnya jangkauan distribusi informasi, sehingga berakibat pada kurangnya minat serta jumlah masyarakat setempat untuk berpartisipasi dalam program patungan hewan kurban.

Hasil survei terhadap 51 masyarakat beragama Islam dari berbagai kota di Indonesia menunjukkan bahwa meskipun telah banyak platform pembelian hewan kurban secara daring, 90,2% responden memilih untuk berkurban dengan membeli atau patungan hewan kurban di masjid terdekat. Sebanyak 88,2% dari mereka menganggap dapat melihat fisik hewan kurban secara langsung penting dilakukan untuk memastikan bahwa hewan kurban sesuai dengan keinginan pribadi maupun memenuhi syarat hewan kurban.

Sejumlah penelitian telah dilakukan untuk meningkatkan pengalaman baik bagi panitia maupun jemaah dalam manajemen pembelian atau patungan hewan kurban melalui

pengembangan sistem teknologi informasi. Namun, belum terdapat penelitian dalam pengembangan aplikasi yang memfasilitasi pembelian dan patungan hewan kurban tanpa mengharuskan jemaah untuk mencari partner patungan secara mandiri, serta memungkinkan mereka untuk dapat melihat fisik hewan kurban mereka secara langsung.

Untuk menyelesaikan permasalahan yang telah dijelaskan, penulis melakukan penelitian untuk mengembangkan aplikasi perangkat bergerak berbasis Android bernama Qurbanku yang menyediakan informasi terbaru serta mempermudah proses manajemen pembelian dan patungan hewan kurban oleh panitia maupun jemaah masjid Al Huda. Android dipilih sebagai basis sistem operasi aplikasi yang akan dikembangkan karena penggunaanya di Indonesia mencapai 89.66% pada rentang Juli 2022 - Juli 2023 (StatCounter Global Stats, 2023) .

Melalui aplikasi ini, panitia kurban dapat melakukan proses pendataan hewan kurban, sementara jemaah dapat mencari lokasi masjid terdekat untuk melihat seluruh informasi, hingga melakukan transaksi pembelian atau patungan hewan kurban secara daring. Dengan demikian, diharapkan aplikasi dapat meningkatkan jumlah dan partisipasi masyarakat sekitar masjid Al Huda dalam berkurban.

Model SDLC *Waterfall* dapat diterapkan ketika seluruh kebutuhan dari perangkat lunak telah dipahami dengan baik, serta memiliki kemungkinan berubah yang sangat kecil (Sommerville, 2011). Oleh karena narasumber dapat memberikan seluruh kebutuhan dan batasan perangkat lunak yang jelas, penulis menerapkan model SDLC *Waterfall* pada pengembangan aplikasi Qurbanku dalam penelitian ini. Melalui pengembangan yang terencana dengan baik, hasil produk aplikasi akan berkualitas tinggi, memiliki seluruh fitur yang dibutuhkan, serta dapat menyelesaikan permasalahan yang diangkat dalam penelitian.

2. LANDASAN KEPUSTAKAAN

2.1 SDLC Waterfall

Model SDLC *Waterfall* adalah model pertama yang diperkenalkan sebagai alur pengembangan perangkat lunak melalui tahap rekayasa sistem yang lebih umum. Oleh karena adanya aliran dari satu tahap ke tahap berikutnya, model ini dikenal dengan nama *Waterfall* atau air terjun. Pada prinsipnya, model SDLC *Waterfall* hanya dapat digunakan apabila seluruh kebutuhan dan persyaratan sistem telah dipahami dengan baik, serta memiliki kemungkinan berubah yang sangat kecil. Meskipun demikian, oleh karena penerapannya yang tergolong mudah, model SDLC *Waterfall* masih banyak digunakan pada proyek pengembangan perangkat lunak (Sommerville, 2011).

2.2 Firebase

Firebase merupakan sebuah layanan yang disediakan Google dengan kategori BaaS (*Backend as a Service*) untuk memudahkan pengembang dalam melakukan konfigurasi sisi *cloud* dan *backend* (Google Android Developer, 2023). Penelitian ini menggunakan beberapa produk dari Firebase, yaitu Cloud Firestore, Cloud Storage, dan Firebase Authentication.

2.3 Pengujian Unit

Pengujian unit merupakan salah satu tipe pengujian dengan pendekatan *white-box testing* yang menguji komponen individual program, seperti fungsi atau metode, dan klas sebagai objek pengujian yang paling sederhana. *Basis path testing* dapat digunakan pada pengujian unit dengan melakukan pengukuran tingkat kompleksitas algoritma hasil perancangan. Langkah-langkah yang dapat dilakukan adalah dengan membuat *flow graph* berdasarkan pemetaan dari struktur algoritma, menentukan ukuran kompleksitas algoritma (*cyclomatic complexity*), lalu diakhiri dengan pendefinisian kasus uji dari algoritma (Sommerville, 2011).

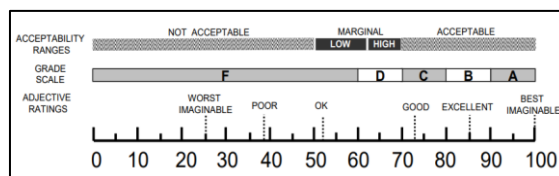
2.4 Pengujian Validasi

Pengujian validasi merupakan pengujian fungsional yang hanya memperhatikan fungsionalitas sistem dan bukan fokus kepada teknis kode implementasi perangkat lunak. Pengujian validasi dapat dilakukan dengan pendekatan *black-box testing*, yaitu pengujian

yang memperlakukan sistem sebagai “kotak hitam”, sehingga perilakunya hanya dapat ditentukan dengan melakukan rekayasa pada pemberian *input* dan *output* yang terkait untuk menguji spesifikasi sistem (Sommerville, 2011).

2.5 System Usability Scale

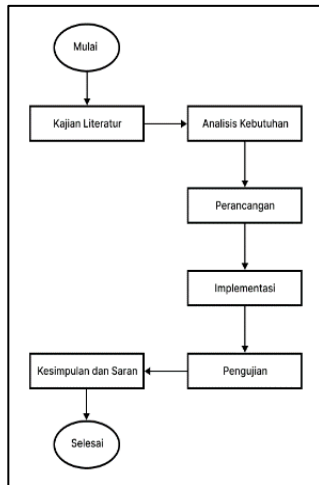
Usabilitas merupakan nilai tentang sejauh mana suatu produk, layanan atau sistem dapat digunakan oleh pengguna tertentu untuk mencapai tujuan yang telah ditetapkan. Penilaian ini mempertimbangkan efektivitas, efisiensi, dan kepuasan dalam konteks penggunaan yang spesifik (ISO 9241-11, 2018). Pengujian usabilitas dilakukan untuk mengukur apakah sebuah perangkat lunak telah terbukti efisien, efektif dan memuaskan sehingga mempermudah pekerjaan pengguna. Nilai usabilitas dari suatu produk dapat diukur menggunakan metode *System Usability Scale* (SUS) dengan mendefinisikan nilai rentang dari sepuluh pertanyaan yang diberikan pada partisipan. Nilai rentang atau yang biasa disebut dengan skala *likert* didefinisikan dari angka satu sampai lima, yaitu nilai satu menyatakan sangat tidak setuju, hingga nilai lima menyatakan sangat setuju. Gambar 1 menunjukkan nilai rentang hasil analisis pada sistem penilaian SUS.



Gambar 1. Rentang Nilai SUS

3. METODOLOGI

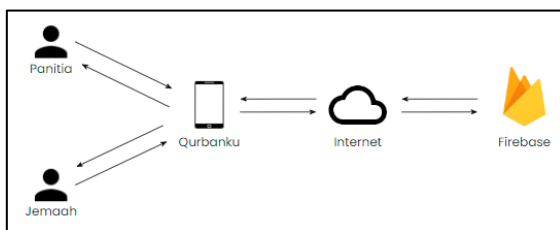
Untuk menghasilkan produk aplikasi berkualitas tinggi, penelitian dilakukan dengan menerapkan tahapan-tahapan terstruktur sesuai ilustrasi pada Gambar 2. Penelitian terbagi menjadi empat tahapan utama, yaitu analisis kebutuhan, perancangan, implementasi, dan pengujian.



Gambar 2. Alur Metode Penelitian

3.1 Analisis Kebutuhan

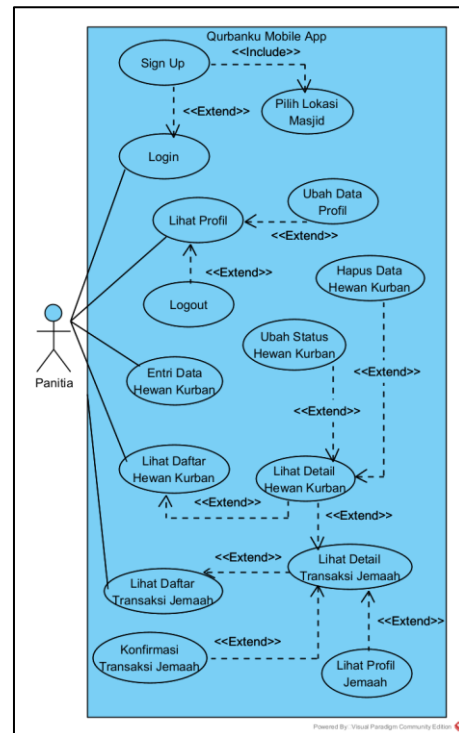
Sistem yang akan dikembangkan pada penelitian ini bernama Qurbanku, yaitu sebuah aplikasi bergerak berjenis *customized software* yang dikembangkan untuk masjid Al Huda, Malang. Penggunaan aplikasi Qurbanku memungkinkan manajemen pendataan hewan kurban oleh panitia kurban, serta transaksi pembelian dan patungan oleh jemaah. Pengelolaan dan penyimpanan data pada aplikasi Qurbanku dilakukan melalui pemanfaatan fitur Firebase Authentication, Cloud Firestore, dan Cloud Storage yang dimiliki oleh Firebase, sehingga mengharuskan pengguna untuk memiliki koneksi internet saat menggunakan aplikasi.



Gambar 3. Gambaran Umum Sistem

Proses analisis kebutuhan diawali dengan wawancara terhadap panitia kurban masjid Al Huda sebagai narasumber. Sementara itu, penulis tidak melakukan proses penggalan data terhadap jemaah masjid Al Huda karena seluruh kebutuhan jemaah dalam proses transaksi dan pencatatan patungan hewan kurban telah didapatkan melalui wawancara dengan panitia kurban masjid Al Huda. Hasil wawancara akan diolah dalam rekayasa kebutuhan melalui identifikasi aktor, pendefinisian kebutuhan fungsional, yang kemudian direpresentasikan

dalam bentuk *usecase diagram*. Melalui proses identifikasi aktor, didapatkan kesimpulan bahwa diperlukan dua aktor untuk dapat menjalankan fungsi aplikasi dengan baik, yaitu aktor panitia dan jemaah. Sementara itu, seluruh fungsionalitas aplikasi yang didapatkan dari hasil pengolahan kebutuhan fungsional masing-masing aktor akan direpresentasikan dalam *usecase diagram*. Gambar 4 merupakan *usecase diagram* milik aktor panitia.



Gambar 4. Usecase Diagram Aplikasi Qurbanku

Setiap *usecase* yang dibuat kemudian diuraikan dalam bentuk skenario. Contoh pembuatan skenario *usecase* akan dijelaskan pada Tabel 1, yaitu pada *usecase* Entri Data Hewan Kurban.

Tabel 1. Skenario Usecase Entri Data Hewan Kurban

Skenario dari <i>usecase</i> Entri Data Hewan Kurban	
Tujuan	Data hewan kurban yang terbaru dapat ditambahkan ke dalam sistem.
Aktor	Panitia
Prakondisi	Halaman tambah data hewan telah terbuka.
Aliran Utama	<ol style="list-style-type: none"> 1. Panitia memilih tanggal hari raya Iduladha. 2. Panitia meminta sistem untuk memproses. 3. Sistem menampilkan halaman entri data hewan kurban dengan opsi pemilihan tanggal hari raya Iduladha.

4. Sistem menampilkan halaman tambah deskripsi hewan kurban dengan mencantumkan formulir entri data hewan kurban yang berisi foto hewan, kolom catatan, serta kolom deskripsi hewan yang berisi nama, jenis, berat, warna, biaya operasional, harga hewan, dan jumlah patungan.
5. Panitia mengisi data pada seluruh kolom formulir.
6. Panitia meminta sistem untuk memproses.
7. Sistem menampilkan halaman detail hewan kurban.

Aliran Alternatif

- 6.a. Jika panitia tidak mengisi salah satu kolom, sistem menampilkan peringatan "Kolom tidak boleh kosong".
- 6.b. Jika panitia belum memilih foto hewan, sistem menampilkan peringatan "Anda belum memilih foto hewan!".
- 6.c. Jika sistem gagal memproses atau koneksi internet tidak aktif, sistem menampilkan pesan kesalahan "Terdapat kesalahan, data hewan gagal ditambahkan".

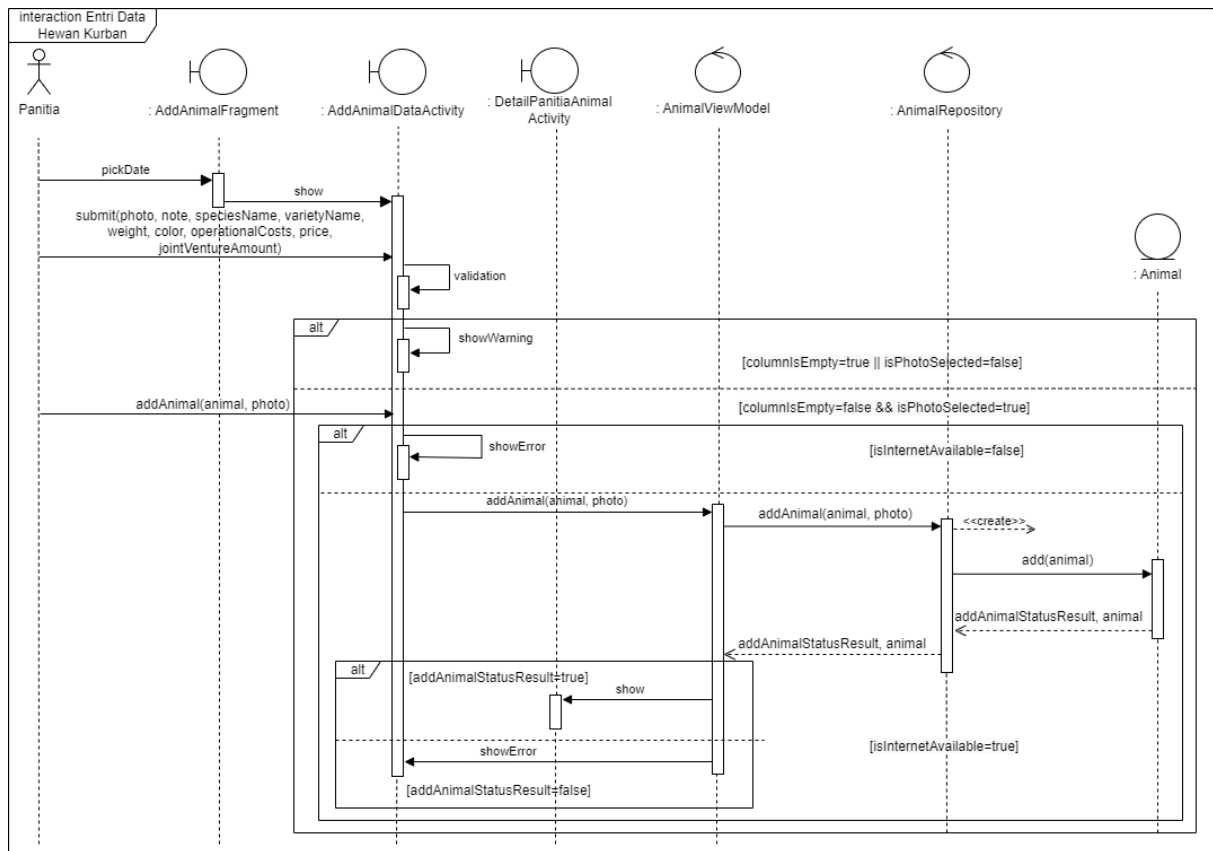
Kondisi Akhir	Halaman detail hewan kurban ditampilkan sistem.
----------------------	---

3.2 Perancangan

Hasil dari tahap analisis kebutuhan yang telah divalidasi kemudian diproses pada tahap perancangan, yaitu terdiri dari perancangan arsitektural, perancangan data, perancangan antarmuka, dan perancangan komponen.

3.2.1 Perancangan Arsitektural

Pada perancangan arsitektural, dilakukan pendefinisian setiap elemen utama dan relasinya dalam bentuk subsistem berdasarkan struktur penyusun perangkat lunak yang dikembangkan. Pada tahap ini, *Model-View-ViewModel (MVVM)* digunakan sebagai pola perancangan dengan mendefinisikan struktur dan elemen klas beserta interaksinya, yaitu melalui pembuatan *sequence diagram* dan *class diagram*. Gambar 5 merupakan contoh pembuatan *sequence diagram* yang didasarkan pada skenario *usecase* Entri Data Hewan Kurban.



Gambar 5. *Sequence Diagram* Entri Data Hewan Kurban

3.2.2 Perancangan Data

Tahap perancangan data dilakukan melalui proses transformasi model domain informasi dari hasil analisis kebutuhan menjadi struktur data yang diperlukan. Struktur basis data pada pengembangan perangkat lunak dalam penelitian ini menerapkan sistem manajemen basis data NoSQL dengan tiga entitas, yaitu user, animal, dan transaction. Setiap pengambilan data pada masing-masing entitas dilakukan dengan mentransformasi ke dalam bentuk skema struktur data JSON yang contohnya dapat dilihat pada Gambar 6, yaitu pada entitas user.

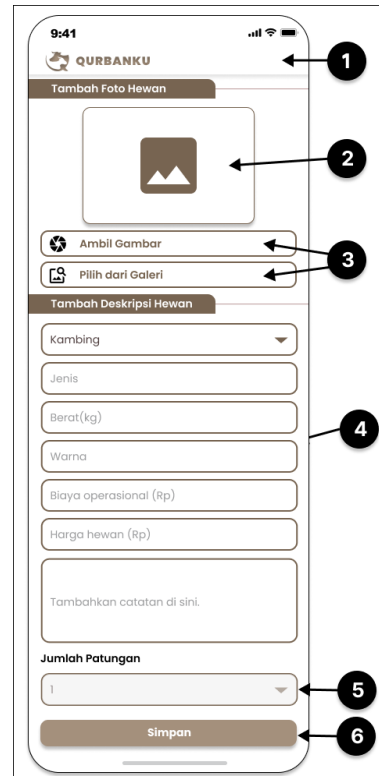
```

object {1}
  user {1}
    octKgo6CfNREHIqksF17e5hs2jF2 {12}
      uid : octKgo6CfNREHIqksF17e5hs2jF2
      email : masjidAlHuda@gmail.com
      name : Al-Huda
      headName : Drs. Slamet Riyadi
      phoneNumber : 082232514210
      admin : true
      address : null
      latitude : -7.941818960519603
      longitude : 112.62731473892927
      bankName : BNI
      bankAccountNumber : 0968116943
      bankAccountName : SYAHRONI
    
```

Gambar 6. Skema Struktur Data JSON Entitas user

3.2.3 Perancangan Antarmuka

Tahap perancangan antarmuka dilakukan untuk mengetahui pola komunikasi suatu perangkat lunak dengan sistem yang ada dalam perangkat lunak itu sendiri, dengan sistem lain, atau bahkan dengan penggunanya, yaitu manusia. Gambar 7 dan Tabel 2 merupakan contoh perancangan antarmuka beserta penjelasannya, yaitu pada usecase Entri Data Hewan Kurban.



Gambar 7. Perancangan Antarmuka Entri Data Hewan Kurban

Tabel 2. Penjelasan Perancangan Antarmuka Entri Data Hewan Kurban

No.	Nama Objek	Jenis Layout	Keterangan
1	Logo Aplikasi	Toolbar	Komponen yang berisi logo dari aplikasi.
2	Container Foto Hewan Kurban	ImageVie w	Container untuk menampilkan foto yang dipilih.
3	Tombol Pilih Gambar	Button	Tombol untuk menampilkan kamera atau galeri untuk memilih foto hewan kurban.
4	Formulir Isi Data Hewan Kurban	EditText	Formulir yang berisi daftar layout EditText yang digunakan untuk mengisi data hewan kurban.
5	Dropdown Jumlah Patungan	Spinner	Komponen untuk menampilkan daftar jumlah patungan melalui fitur dropdown.
6	Tombol Simpan Data Hewan	Button	Tombol untuk menyimpan dan mengirim data hewan kurban ke

Kurban	dalam basis data.
--------	-------------------

3.2.4 Perancangan Komponen

Tahap perancangan komponen dalam pengembangan perangkat lunak dilakukan untuk mendekomposisi sub-sistem menjadi komponen yang lebih detail. Perancangan komponen pada penelitian dilakukan dengan pembuatan *pseudocode* sebagai gambaran algoritma dari setiap fungsionalitas yang akan dikembangkan.

3.3 Implementasi

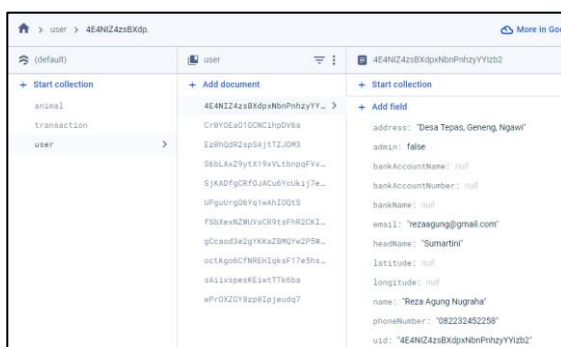
Tahap implementasi dilakukan dengan pengkodean hasil perancangan yang dideskripsikan melalui beberapa subbab, mulai dari penjelasan spesifikasi lingkungan pengembangan, implementasi data, implementasi antarmuka, dan implementasi kode program. Pengkodean dilakukan pada *Integrated Development Environment (IDE)* Android Studio dengan bahasa pemrograman Kotlin untuk menjadi produk aplikasi perangkat bergerak.

3.3.1 Spesifikasi Lingkungan Pengembangan Perangkat Lunak

Spesifikasi lingkungan pengembangan perangkat lunak akan didefinisikan pada tiga kategori, yaitu spesifikasi perangkat keras pengembang (*Development Hardware*), spesifikasi perangkat keras perangkat bergerak (*Mobile Hardware*), dan spesifikasi perangkat lunak (*Software*).

3.3.2 Implementasi Data

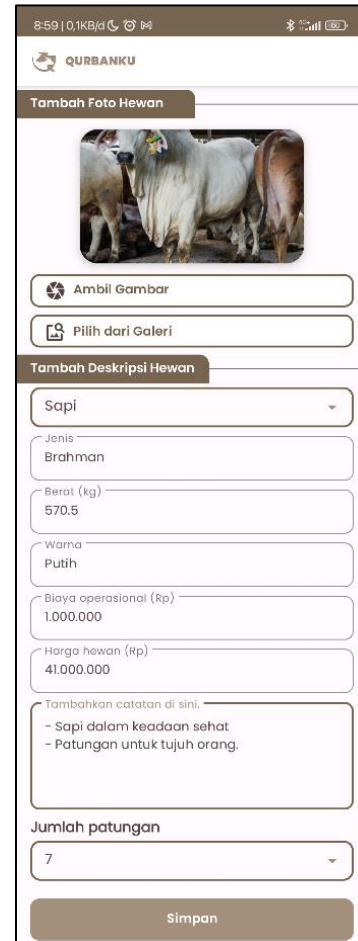
Implementasi data dilakukan pada setiap entitas yang telah dibuat dalam tahap perancangan, yaitu *user*, *animal*, dan *transaction* dengan mentransformasikan ke dalam bentuk dokumen *Firestore*. Contoh implementasi data dapat dilihat pada Gambar 8, yaitu pada entitas *user*.



Gambar 8. Implementasi Data Entitas *user*

3.3.3 Implementasi Antarmuka

Implementasi antarmuka dilakukan berdasarkan hasil perancangan antarmuka aplikasi *Qurbanku*. Gambar 9 merupakan contoh implementasi antarmuka, yaitu pada *usecase* Entri Data Hewan Kurban.



Gambar 9. Implementasi Antarmuka Entri Data Hewan Kurban

Antarmuka Entri Data Hewan Kurban digunakan oleh panitia untuk menambahkan hewan kurban yang disediakan pada program pembelian maupun patungan. Panitia diharuskan untuk mengisi seluruh data pada formulir yang berkaitan dengan informasi hewan kurban, mulai dari foto, deskripsi hewan, hingga jumlah patungan. Setelah berhasil mengisi data hewan kurban, panitia dapat menekan tombol "Simpan" agar sistem dapat memproses penambahan data hewan kurban baru.

3.3.4 Implementasi Kode Program

Implementasi kode program dilakukan menggunakan bahasa pemrograman Kotlin

berdasarkan *pseudocode* hasil tahap perancangan komponen melalui IDE Android Studio yang terhubung dengan platform Firebase. Konfigurasi yang dilakukan pada fase ini akan memungkinkan pengguna yang memiliki perangkat Android dengan versi minimal 5.0 Lollipop untuk menjalankan aplikasi.

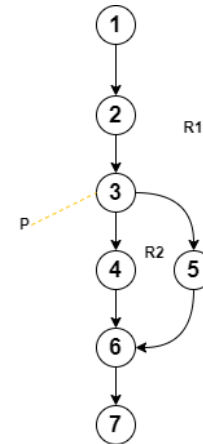
3.4 Pengujian

Tahap pengujian perangkat lunak dilakukan untuk memastikan bahwa produk aplikasi layak digunakan, dapat menjalankan semua fungsionalitas yang diperlukan, serta untuk menemukan kecacatan sebelum akhirnya digunakan oleh pengguna akhir. Terdapat dua jenis pengujian yang dilakukan, yaitu pengujian fungsional dan pengujian usabilitas. Pengujian fungsional dilakukan dengan dua teknik pengujian, yaitu *white-box testing* melalui pengujian unit, serta *black-box testing* melalui pengujian validasi. Sementara itu, pengujian usabilitas dilakukan dengan metode *System Usability Scale (SUS)*.

3.4.1 Pengujian Unit

Pengujian unit dengan metode *basis path testing* dilakukan terhadap enam *pseudocode* fungsi atau algoritma dari kedua aktor pada perancangan komponen. Pengujian dilakukan dengan melakukan perhitungan nilai *cyclomatic complexity* berdasarkan struktur algoritma yang terdefinisi melalui pembuatan *flow graph*, kemudian diakhiri dengan pendefinisian kasus uji dari algoritma.

Sebagai contoh, dilakukan pengujian terhadap *pseudocode* fungsi `addAnimal` yang terdapat pada fungsionalitas Entri Data Hewan Kurban di lapisan *View*. Melalui pendefinisian struktur proses dalam algoritma, *flow graph* fungsi `addAnimal` dapat dibuat, yaitu pada Gambar 10.



Gambar 10. *Flow Graph* Fungsi `addAnimal` – Lapisan *View*

Berdasarkan *flow graph* yang telah dibuat, nilai *cyclomatic complexity pseudocode* fungsi `addAnimal` dapat dihitung, yaitu sebagai berikut:

- $V(G) = E - N + 2$
 $= 7 - 7 + 2$
 $= 2$
- $V(G) = \text{Jumlah Region}$
 $= 2$
- $V(G) = \text{Jumlah Predicate Node} + 1$
 $= 1 + 1$
 $= 2$

Melalui perhitungan nilai *cyclomatic complexity* yang telah dilakukan, didapatkan kesimpulan bahwa terdapat dua jalur dasar yang dapat didefinisikan. Kasus uji setiap jalur yang terdapat pada algoritma fungsi akan diuraikan pada Tabel 3.

- Jalur 1: 1–2–3–4–6–7
- Jalur 2: 1–2–3–5–6–7

Tabel 3. Kasus Uji Pengujian Unit Fungsi `addAnimal` – Lapisan *View*

No.	Kasus Uji	Target Hasil Akhir	Status Hasil Aktual
1	Nilai <code>isSuccess</code> adalah <code>true</code> .	Memanggil fungsi <code>showDialog</code> dengan mengirim <code>isSuccess</code> dan <code>animal</code> sebagai parameter untuk menampilkan pesan keberhasilan.	Sukses
2	Nilai <code>isSuccess</code> adalah <code>false</code> .	Memanggil fungsi <code>showDialog</code> dengan mengirim <code>false</code> dan <code>null</code> sebagai parameter untuk menampilkan	Sukses

pesan kegagalan.

Melalui perhitungan skor pengujian, diketahui bahwa skor kesesuaian seluruh kasus uji adalah 100%, sehingga membuktikan bahwa seluruh *pseudocode* fungsi atau algoritma hasil perancangan telah sesuai dengan jalur dasar dari *basis path testing*.

3.4.2 Pengujian Validasi

Pengujian validasi dengan pendekatan *black-box testing* dilakukan terhadap seluruh *usecase* yang menggambarkan kebutuhan fungsional dari kedua aktor pada aplikasi Qurbanku. Kasus uji yang diberikan akan disesuaikan dengan aliran yang dimiliki oleh masing-masing *usecase*, kemudian hasil uji aktual akan dibandingkan dengan target hasil uji. Hasil pengujian validasi pada seluruh fungsionalitas aplikasi Qurbanku adalah 100% valid, sehingga membuktikan bahwa pengguna dapat mencapai tujuannya saat menggunakan aplikasi. Contoh pengujian validasi dapat dilihat pada Tabel 4, yaitu pada skenario fungsionalitas Entri Data Hewan Kurban.

Tabel 4. Pengujian Validasi Entri Data Hewan Kurban

Fungsionalitas Entri Data Hewan Kurban			
No.	Kasus Uji	Target Hasil Akhir	Status Hasil Aktual
1	Aktor mengisi seluruh data pada formulir entri data hewan kurban, yaitu foto hewan, kolom catatan, serta daftar kolom deskripsi hewan yang berisi nama, jenis, berat, warna, biaya operasional, harga hewan, dan jumlah patungan.	Sistem berhasil menambah data hewan kurban, lalu menampilkan halaman detail hewan kurban.	Sukses
2	Aktor tidak mengisi salah satu kolom pada formulir entri data hewan kurban.	Sistem menampilkan peringatan "Kolom tidak boleh kosong".	Sukses
3	Aktor tidak memilih foto hewan.	Sistem menampilkan peringatan "Anda belum memilih foto hewan!".	Sukses

4	Sistem gagal memproses karena <i>smartphone</i> aktor tidak memiliki koneksi internet.	Sistem menampilkan pesan kesalahan "Terdapat kesalahan, data hewan gagal ditambahkan".	Sukses
---	--	--	--------

3.4.3 Pengujian Usabilitas

Pengujian usabilitas dilakukan kepada sepuluh partisipan, yaitu masing-masing sebanyak lima partisipan dari pihak panitia kurban maupun pihak jemaah masjid Al Huda yang merepresentasikan aktor untuk mengetahui tingkat kepuasan pengguna aplikasi Qurbanku. Pada pengujian usabilitas, setiap partisipan akan diberikan tugas berbentuk skenario yang mencakup enam fitur utama dalam aplikasi, kemudian diberikan sepuluh pertanyaan SUS untuk dijawab secara langsung. Berdasarkan hasil analisis, didapatkan nilai rata-rata skor SUS sebesar 81,75 sehingga memiliki predikat B atau masuk dalam kategori "*excellent-acceptable*". Hal ini menunjukkan bahwa aplikasi Qurbanku terbukti efektif, efisien, serta memenuhi syarat kepuasan pengguna.

4. PENUTUP

4.1 Kesimpulan

Melalui beberapa tahapan penelitian yang telah dilakukan dengan mengembangkan aplikasi Qurbanku, penulis dapat menarik beberapa kesimpulan sebagai berikut:

1. Berdasarkan hasil tahap analisis kebutuhan, pengembangan aplikasi ditujukan untuk dua tipe aktor, yaitu panitia kurban dan jemaah. Dalam rangka mempermudah penulis untuk memahami sistem, sebanyak 28 fungsionalitas direpresentasikan dalam bentuk *usecase*, sehingga dapat dipetakan dalam sebuah diagram *usecase* beserta masing-masing skenarionya.
2. Perancangan dengan pola MVVM menghasilkan *sequence diagram* dan *class diagram* yang mendefinisikan struktur dan elemen klas beserta interaksinya. Perancangan data menghasilkan entitas *user*, *animal* dan *transaction* dengan skema JSON. Perancangan antarmuka dibuat

- dengan penjelasan antar komponen pada *layout* penyusun, sedangkan perancangan komponen menghasilkan *pseudocode*.
- Implementasi dilakukan berdasarkan hasil rancangan aplikasi menggunakan IDE Android Studio yang terhubung dengan Firebase. Implementasi data menghasilkan tiga *collection* pada Firebase Firestore, yaitu `user`, `animal` dan `transaction`. Implementasi antarmuka dan kode program melalui pengkodean menggunakan bahasa pemrograman Kotlin menghasilkan produk aplikasi Qurbanku.
 - Pengujian fungsional dilakukan dengan dua teknik pengujian, yaitu *white-box testing* melalui pengujian unit, serta *black-box testing* melalui pengujian validasi. Pengujian unit dengan metode *basis path testing* dilakukan dengan menghitung cyclomatic complexity, sehingga membuktikan bahwa seluruh algoritma hasil perancangan telah sesuai dengan jalur dasar. Sementara itu, pengujian validasi mendapatkan hasil 100% valid, membuktikan bahwa pengguna dapat mencapai tujuannya saat menggunakan aplikasi.
 - Pengujian usabilitas menghasilkan skor rata-rata SUS sebesar 81,75 atau masuk dalam kategori “*excellent-acceptable*”, menunjukkan bahwa aplikasi Qurbanku terbukti efektif, efisien, serta memuaskan pengguna dalam mencapai tujuannya.
 - Penggunaan warna tampilan aplikasi dibuat lebih kontras untuk mempermudah pengguna di kalangan usia tua dalam melihat konten dan informasi dalam aplikasi.

5. DAFTAR PUSTAKA

- Google Android Developer, 2023. *Learn more about Android and Firebase*. [Online] Available at: <https://firebase.google.com/docs/android/learn-more> [Accessed 5 September 2023].
- ISO 9241-11, 2018. *Ergonomics of human-system interaction-Part 11: Usability: Definitions and concepts*. [Online] Available at: <https://www.iso.org/obp/ui/#iso:std:iso:9241:-11:ed-2:v1:en> [Accessed 3 September 2023].
- Sommerville, I., 2011. *Software Engineering, 9/E*. Pearson Education India: s.n.
- StatCounter Global Stats, 2023. *Mobile Operating System Market Share Indonesia*. [Online] Available at: <https://gs.statcounter.com/os-market-share/mobile/indonesia> [Accessed 24 August 2023].
- Wardi, S. & Waldi, I., 2021. DISKURSUS PENGAMALAN BERQURBAN MENURUT LDII DAN AL WASHLIYAH; BERQURBAN SECARA BERJAMA’AH (PATUNGAN). *Islamijah: Journal of Islamic Social Sciences* Wardi, Syah, 2(1), p. 19.

4.2 Saran

Berikut merupakan tiga saran dari penulis kepada peneliti yang akan melanjutkan penelitian ini:

- Penggalian data berupa wawancara sebaiknya melibatkan seluruh *stakeholder* yang merupakan aktor dalam aplikasi.
- Peneliti menambahkan fitur notifikasi untuk seluruh fungsionalitas yang berkaitan dengan transaksi.
- Peneliti menambahkan fitur tabungan kurban untuk meningkatkan frekuensi penggunaan aplikasi.