

Desain dan Implementasi Sistem Pengingat Pembayaran Zakat serta Pop-up Edukatif dalam Aplikasi Kalkulator Zakat

Dinda Rahma
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
dindarahma@student.telkomuniversity.
ac.id

Anggunmeka Luhur Prasasti
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
anggunmeka@telkomuniversity.ac.id

Astri Novianty
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
astrinov@telkomuniversity.ac.id

Abstrak — Perkembangan teknologi, khususnya kecerdasan buatan (AI), telah mengubah cara pengelolaan kewajiban agama seperti zakat dan pembagian harta waris. Penelitian ini bertujuan mengembangkan aplikasi AI yang mampu memberikan solusi efisien dan akurat dalam perhitungan zakat dan waris sesuai dengan hukum Islam. Dengan menggunakan pendekatan kualitatif dan metode pengembangan, penelitian ini melalui tahapan analisis kebutuhan, perancangan, pengembangan, pengujian, dan implementasi aplikasi. Hasil penelitian menunjukkan bahwa aplikasi yang dikembangkan memiliki desain UI/UX yang intuitif dan mudah digunakan, serta mampu menghitung zakat dan waris dengan tingkat keakuratan yang tinggi. Pengujian lebih lanjut mengungkapkan bahwa meskipun aplikasi ini berhasil dalam sebagian besar pengujian, terdapat beberapa tantangan dalam usability dan penerimaan pengguna yang perlu diatasi, terutama untuk kelompok pengguna yang lebih tua. Kesimpulannya, teknologi AI dapat berperan signifikan dalam memfasilitasi pengelolaan kewajiban agama, namun perhatian khusus perlu diberikan pada aspek desain, edukasi pengguna, serta pengujian berkelanjutan untuk memastikan penerapannya secara efektif.

Kata kunci— kecerdasan buatan, zakat, waris, UI/UX, Dart

I. PENDAHULUAN

Perkembangan teknologi, khususnya kecerdasan buatan (AI), telah memberikan kontribusi signifikan dalam pengelolaan kewajiban agama, seperti zakat dan pembagian harta waris. Aplikasi berbasis AI mampu memberikan solusi yang efisien dan akurat untuk perhitungan kewajiban tersebut, meskipun tantangan masih ada, terutama terkait dengan kompleksitas hukum waris Islam dan pemahaman pengguna [1][2]. Salah satu tantangan utama adalah kompleksitas aturan hukum waris yang berbeda di berbagai negara dan situasi keluarga, yang memerlukan pengetahuan mendalam dalam pengembangan aplikasi untuk memastikan akurasi dan keabsahan perhitungan [3][4].

Desain antarmuka pengguna (UI/UX) juga merupakan faktor penting untuk keberhasilan aplikasi. Antarmuka yang intuitif membantu pengguna memahami konsep dan membuat keputusan yang lebih baik, namun istilah-istilah hukum yang kompleks masih dapat membingungkan pengguna awam, menunjukkan kebutuhan akan panduan yang lebih komprehensif [5][6]. Pengembangan aplikasi ini memerlukan pendekatan multidisipliner yang

melibatkan ahli komputer dan ilmu agama untuk menyeimbangkan aspek teknis dan hukum Islam [10]. Selain itu, edukasi pengguna juga penting, dengan menyediakan sumber daya pendidikan yang membantu meningkatkan pemahaman pengguna tentang zakat dan hukum waris [8].

Meskipun sudah banyak aplikasi serupa, masih terdapat kekurangan, seperti antarmuka yang kurang menarik dan kurangnya akurasi dalam perhitungan waris yang kompleks. Penggunaan AI, khususnya sistem pakar, menawarkan solusi potensial dengan meningkatkan transparansi, efisiensi, dan adaptasi dalam perhitungan, serta memberikan hasil yang lebih akurat dan andal [11].

II. KAJIAN TEORI

A. Zakat dan Waris

Zakat dan waris merupakan dua pilar penting dalam syariat Islam yang mengatur aspek sosial dan ekonomi umat Muslim. Zakat, salah satu dari lima rukun Islam, wajib ditunaikan oleh setiap Muslim yang memiliki harta mencapai nisab dan haul. Zakat tidak hanya berfungsi sebagai ibadah, tetapi juga sebagai instrumen sosial-ekonomi untuk mengurangi kesenjangan sosial dan membantu mereka yang membutuhkan. Hukum zakat menekankan kewajiban untuk mendistribusikan kekayaan kepada yang berhak, dengan besaran zakat umumnya sebesar 2,5% dari harta yang telah memenuhi syarat. Konsep ini mengedepankan pentingnya menjaga sirkulasi kekayaan dalam masyarakat Muslim agar tercipta keadilan sosial [12].

Di sisi lain, hukum waris dalam Islam diatur secara rinci dalam Al-Qur'an, sunnah, dan ijma' ulama. Prinsip waris Islam bertujuan untuk memastikan keadilan dalam distribusi harta peninggalan, di mana hak waris dibagi secara adil sesuai kedudukan keluarga. Sistem kewarisan ini dikenal dengan prinsip bilateral, yang memungkinkan seseorang mewarisi harta dari kedua jalur, baik ayah maupun ibu. Terdapat tiga kelompok utama dalam sistem pewarisan, yaitu ashabul furudh, ashobah, dan dzawil arham, yang masing-masing memiliki peran dalam pembagian warisan sesuai dengan ketentuan syariat [13].

Kedua aspek ini, zakat dan waris, mencerminkan pentingnya keseimbangan antara keadilan sosial dan tanggung jawab finansial dalam Islam. Zakat membersihkan

harta dan jiwa, serta menjaga kesejahteraan sosial, sedangkan hukum waris memastikan kelanjutan kesejahteraan keluarga melalui pembagian harta yang adil dan terstruktur [8] [3].

B. Bahasa Pemrograman

Bahasa pemrograman adalah alat dasar yang digunakan oleh pengembang perangkat lunak untuk memberi instruksi pada komputer agar dapat melakukan tugas tertentu [12]. Setiap bahasa pemrograman memiliki kelebihan dan kekurangannya sendiri, yang membuatnya lebih cocok untuk jenis proyek tertentu. Dua bahasa pemrograman yang populer adalah Dart dan Kotlin, yang keduanya menawarkan fitur-fitur khusus yang mendukung pengembangan aplikasi modern. Dart adalah bahasa pemrograman yang dikembangkan oleh Google dan dirancang khusus untuk pengembangan aplikasi yang responsif dan efisien. Salah satu kelebihan utama Dart adalah kemampuannya untuk menghasilkan aplikasi dengan kinerja tinggi, terutama dalam lingkungan pengembangan web. Dart menggunakan paradigma *Object-Oriented Programming* (OOP), yang membantu dalam pengorganisasian dan pengelolaan kode, sehingga memudahkan pengembang yang sudah terbiasa dengan bahasa pemrograman lain untuk beralih atau mengintegrasikannya dengan aplikasi yang ada. Namun, salah satu kekurangan Dart adalah adopsi yang relatif rendah di luar ekosistem Google, yang membuatnya kurang fleksibel untuk pengembangan aplikasi umum yang memerlukan sifat multiguna (Ray, 2021).

Selain itu, Kotlin adalah bahasa pemrograman yang semakin populer terutama dalam pengembangan aplikasi Android. Salah satu keunggulan Kotlin adalah sintaksisnya yang ringkas dan sederhana, yang memudahkan pengembang dalam menulis dan memelihara kode. Selain itu, Kotlin mendukung interoperabilitas penuh dengan Java, yang memungkinkan pengembang untuk menggunakan pustaka dan framework Java yang ada, membuatnya menjadi pilihan yang kuat dalam pengembangan aplikasi mobile. Namun, sebagai bahasa yang relatif baru, Kotlin masih menghadapi tantangan dalam hal dukungan komunitas dan ketersediaan sumber daya, seperti tutorial dan dokumentasi, yang mungkin belum sebanding dengan bahasa yang lebih matang [12].

Selain bahasa pemrograman, pemilihan *Integrated Development Environment* (IDE) juga merupakan faktor penting dalam pengembangan perangkat lunak. *Android Studio* dan *Visual Studio Code* (VS Code) adalah dua IDE populer yang sering digunakan oleh pengembang. *Android Studio*, yang dikembangkan oleh Google, menyediakan alat-alat yang lengkap untuk pengembangan aplikasi Android, termasuk emulator, debugger, dan editor kode. Namun, *Android Studio* membutuhkan sumber daya yang cukup besar, terutama dalam hal memori dan CPU, yang dapat menjadi hambatan bagi pengembang dengan perangkat keras yang kurang bertenaga. VS Code, di sisi lain, dirancang untuk menjadi ringan dan cepat, membuatnya responsif bahkan pada komputer dengan spesifikasi lebih rendah. VS Code juga memiliki banyak ekstensi yang memungkinkan dukungan untuk berbagai bahasa pemrograman dan framework, meskipun tidak menyediakan *Visual Designer* sekomprensif *Android Studio* [13].

III. METODE

Penelitian ini menggunakan pendekatan kualitatif dengan jenis penelitian pengembangan untuk merancang aplikasi berbasis sistem pakar yang secara otomatis

menghitung zakat dan waris sesuai dengan aturan Islam. Proses dimulai dengan analisis kebutuhan, di mana peneliti mengidentifikasi fitur-fitur yang diperlukan melalui studi literatur dan wawancara dengan ahli di bidang zakat dan waris. Tahap ini bertujuan untuk memastikan bahwa aplikasi yang dikembangkan memenuhi harapan pengguna dan sesuai dengan kebutuhan mereka. Setelah kebutuhan ditentukan, perancangan aplikasi dilakukan dengan menggunakan metode *rule-based expert system*, yang memungkinkan aplikasi untuk beroperasi berdasarkan aturan-aturan yang telah ditetapkan. Perancangan ini mencakup arsitektur sistem serta desain antarmuka yang intuitif, memastikan bahwa aplikasi dapat menjalankan fungsinya dengan efektif.

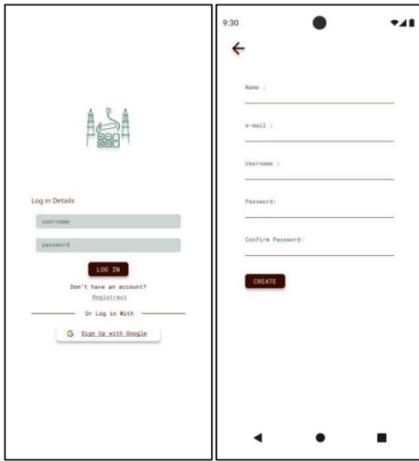
Tahap berikutnya adalah pengembangan, di mana desain yang telah dirancang diimplementasikan menjadi aplikasi fungsional. Proses ini melibatkan pemrograman serta pengujian fitur untuk memastikan bahwa aplikasi berjalan sesuai dengan harapan. Pengujian dilakukan menggunakan data uji untuk mengevaluasi akurasi perhitungan zakat dan waris, yang hasilnya dibandingkan dengan perhitungan manual atau oleh ahli. Setelah pengujian dan evaluasi, aplikasi diimplementasikan dengan pelatihan dan panduan penggunaan untuk pengguna. Penelitian ini juga mencakup tahap pemeliharaan dan pengembangan lebih lanjut untuk terus memantau kinerja aplikasi, melakukan perbaikan, dan memastikan relevansi serta efektivitas aplikasi di masa depan.

IV. HASIL DAN PEMBAHASAN

A. UI/UX Tampilan Aplikasi Rancangan

User Interface (UI) dan *User Experience* (UX) adalah elemen kritis dalam pengembangan aplikasi, terutama dalam memastikan bahwa pengguna dapat berinteraksi dengan sistem secara efisien dan menyenangkan. UI berkaitan dengan aspek visual aplikasi, seperti tata letak, warna, tipografi, dan ikon, yang dirancang untuk menciptakan antarmuka yang menarik secara estetis. Sementara itu, UX fokus pada keseluruhan pengalaman pengguna saat berinteraksi dengan aplikasi, termasuk kemudahan navigasi, kecepatan respons, dan kepuasan pengguna dalam mencapai tujuan mereka melalui aplikasi tersebut. Dalam merancang tampilan aplikasi berbasis sistem pakar untuk menghitung zakat dan waris, prinsip-prinsip UI/UX perlu diterapkan dengan cermat. Pertama, desain harus intuitif, dengan elemen-elemen yang mudah dikenali dan digunakan oleh pengguna dari berbagai latar belakang. Misalnya, tombol-tombol dan menu harus diberi label yang jelas dan diletakkan di lokasi yang mudah dijangkau. Selain itu, penting untuk memastikan bahwa alur kerja aplikasi logis dan mendukung pengguna dalam menyelesaikan tugas-tugas mereka tanpa hambatan yang tidak perlu [14].

Gambar 1 menunjukkan tampilan login dan registrasi aplikasi. Pada layar login, pengguna perlu memasukkan username dan password yang telah mereka daftarkan sebelumnya. Alternatifnya, pengguna juga bisa memilih untuk masuk menggunakan akun Google mereka. Jika pengguna ingin mendaftar untuk membuat akun baru, mereka akan diminta untuk mengisi informasi seperti nama, email, username, dan password pada bagian registrasi.

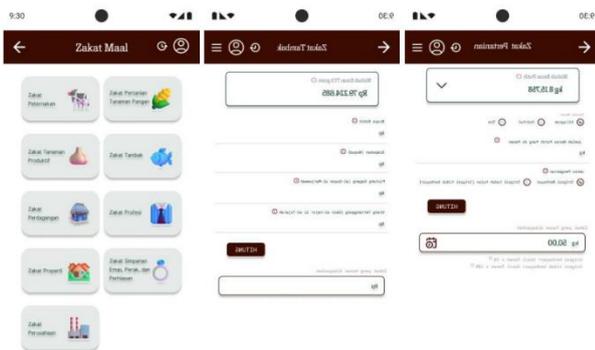


Gambar 1. Tampilan Login dan Registrasi Aplikasi (Rahma dkk, 2024)



Gambar 2. Daily Pop-up dan Home Page Aplikasi (Rahma dkk, 2024)

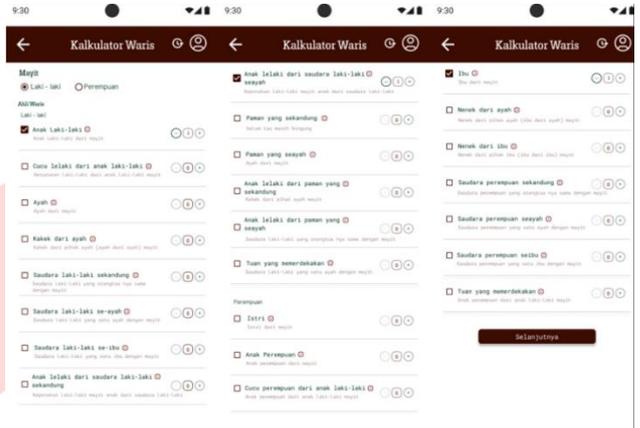
Gambar 2 menunjukkan halaman utama, di mana setelah pengguna berhasil masuk, mereka akan melihat pop-up harian yang berisi materi tentang ilmu zakat dan waris. Jika pop-up harian tersebut ditutup, pengguna akan diarahkan ke halaman utama yang menampilkan opsi menu untuk zakat atau waris.



Gambar 3. Tampilan Zakat Maal (Rahma dkk, 2024)

Gambar 3 menunjukkan berbagai pilihan jenis zakat maal yang dapat dihitung oleh pengguna, termasuk zakat peternakan, pertanian tanaman pangan, tanaman produktif, tambak, profesi, perdagangan, properti, simpanan emas, perak, perhiasan, dan zakat perusahaan. Pada Menu Zakat Tambak, pengguna diminta untuk memasukkan nilai-nilai

seperti biaya bibit, simpanan (nuqud), piutang dagang (al-duyun al-marjuwah), dan utang tertanggung (dain al-tajir ila al-tijarah). Setelah menekan tombol "Hitung," jumlah zakat yang harus dibayarkan akan muncul. Di Menu Zakat Pertanian, pengguna dapat memasukkan hasil panen dalam satuan berat yang berbeda (kilogram, kuintal, ton) dan memilih jenis pengairan, apakah berbayar atau tadah hujan. Setelah menekan tombol "Hitung," jumlah zakat yang harus dibayar akan ditampilkan berdasarkan jenis pengairan yang dipilih.



Gambar 4. Tampilan Input pada Fitur Waris (Rahma dkk, 2024)

Gambar 4 menunjukkan tampilan input pada fitur waris. Pengguna diminta untuk memasukkan data yang diperlukan guna proses perhitungan zakat atau waris. Langkah pertama adalah memilih jenis kelamin jenazah, yang akan mempengaruhi hasil perhitungan karena aturan pembagian harta waris bisa berbeda untuk pria dan wanita. Setelah itu, pengguna perlu memasukkan informasi mengenai ahli waris yang ditinggalkan. Data yang harus diisi mencakup nama setiap ahli waris dan jumlah bagian warisan yang diterima oleh masing-masing. Informasi ini sangat penting untuk memastikan bahwa pembagian warisan dilakukan secara adil dan sesuai dengan ketentuan hukum yang berlaku. Pada tahap ini, pengguna harus memastikan bahwa semua data yang dimasukkan sudah benar dan lengkap untuk mendapatkan hasil perhitungan yang akurat. Setelah semua informasi telah diisi, pengguna dapat menekan tombol "Selanjutnya" untuk melanjutkan ke halaman berikutnya. Halaman ini akan menampilkan hasil perhitungan berdasarkan data yang telah dimasukkan. Proses ini membantu pengguna memahami bagaimana harta warisan akan dibagi sesuai dengan ketentuan yang berlaku, memastikan bahwa pembagian dilakukan dengan tepat dan adil sesuai dengan prinsip-prinsip hukum waris.

B. Source Code

Source code, atau kode sumber, adalah sekumpulan instruksi tertulis yang membentuk program komputer dan ditulis dalam bahasa pemrograman tertentu. Kode ini berfungsi sebagai cetak biru yang mengarahkan komputer dalam menjalankan tugas tertentu, mulai dari operasi dasar hingga fungsi kompleks. Setiap baris dalam source code menyertakan perintah yang memberi tahu komputer apa yang harus dilakukan dan bagaimana melakukannya. Bahasa pemrograman seperti Python, Java, atau C++ digunakan untuk menulis source code, masing-masing dengan sintaks dan struktur yang berbeda. Pada umumnya, source code dikembangkan oleh programmer dengan menuliskan

serangkaian perintah logis yang harus diikuti oleh mesin. Kode ini harus ditulis dengan presisi tinggi karena kesalahan kecil dapat menyebabkan program tidak berfungsi dengan baik atau mengalami error. Setelah source code selesai ditulis, kode tersebut biasanya dikompilasi atau diterjemahkan menjadi bahasa mesin yang dapat dipahami dan dijalankan oleh komputer. Proses ini memungkinkan komputer untuk mengeksekusi instruksi yang telah ditulis dalam kode sumber.

Login

```
Future < void > signInWithGoogle() async
{ try { final GoogleSignInAccount?
googleUser = await
GoogleSignIn().signIn();
if (googleUser != null) { final
GoogleSignInAuthentication googleAuth =
await googleUser.authentication;
final AuthCredential credential =
GoogleAuthProvider.credential(
accessToken : googleAuth.accessToken,
idToken : googleAuth.idToken,
);
```

Fungsi `signInWithGoogle()` dalam kode di atas mengimplementasikan fitur autentikasi menggunakan akun Google dalam aplikasi. Pertama, fungsi ini memulai proses login Google dengan meminta pengguna untuk masuk menggunakan akun Google mereka. Jika pengguna berhasil melakukan autentikasi, aplikasi akan mendapatkan objek `GoogleSignInAuthentication` yang berisi token akses dan ID token. Token ini kemudian digunakan untuk membuat objek `AuthCredential` yang diperlukan untuk autentikasi lebih lanjut. Fitur ini memungkinkan pengguna untuk masuk ke aplikasi dengan mudah menggunakan akun Google mereka, selain metode tradisional yang memerlukan username dan password. Dengan mendukung autentikasi Google, aplikasi memberikan opsi yang lebih aman dan nyaman bagi pengguna untuk mengakses halaman utama aplikasi, sambil tetap menawarkan kemampuan untuk mengingat informasi login dan memulihkan kata sandi jika diperlukan.

Daily Pop-Up

```
void main() {
  runApp(MaterialApp(
    home: Scaffold(
      body: Center(
        child: ElevatedButton(
          onPressed: () {
            showDialog(
              context: context,
              builder: (BuildContext context) {
                return DialogBoxScreen();
              },
            );
          },
          child: Text('Tampilkan Dialog'),
        ),
      ),
    ),
  );
```

```
));
```

Kode di atas mendemonstrasikan implementasi fitur pop-up harian dalam aplikasi Flutter yang menampilkan tips mengenai ilmu waris dan zakat dalam Islam. Dalam fungsi `main()`, aplikasi dijalankan menggunakan `MaterialApp` yang berisi `Scaffold` dengan `Center` dan `ElevatedButton`. Ketika tombol ditekan, sebuah dialog muncul menggunakan `showDialog()`, yang menampilkan widget `DialogBoxScreen()`. Dialog ini menampilkan tips secara acak dari daftar yang telah disiapkan, dengan tampilan yang disesuaikan dengan sudut yang dibulatkan dan dilengkapi ikon lampu di bagian atas sebagai penanda bahwa itu adalah saran. Pengguna dapat menutup dialog dengan menekan tombol "Tutup". Implementasi ini sangat berguna dalam aplikasi edukasi yang bertujuan memberikan informasi dan inspirasi harian kepada pengguna tentang ilmu waris dan zakat.

Reminder

```
Future<void> _scheduleNotification()
async {
  final DateTime reminderDateTime =
    DateTime(
      selectedDate.year,
      selectedDate.month,
      selectedDate.day,
      selectedTime.hour,
      selectedTime.minute,
    );

  const String title = 'Peringat
Pembayaran';
  const String body = 'Jangan lupa
untuk melakukan pembayaran zakat!';

  print('Scheduling notification for:
$reminderDateTime');
  await NotificationService()
    .scheduleNotification(reminderDateTime,
    title, body);

  ScaffoldMessenger.of(context).showSnackBar(
    const SnackBar(content:
    Text('Peringat telah dijadwalkan')),
  );
}
```

Zakat Fitrah

```
class ZakatFitrahPage extends
StatefulWidget {
  @override
  _ZakatFitrahPageState createState() =>
  _ZakatFitrahPageState();
}

class _ZakatFitrahPageState extends
State<ZakatFitrahPage>
```

```

{
  TextEditingController
  _necessitiesController =
TextEditingController();
  TextEditingController
  _pricePerKgController =
TextEditingController();
  double _zakatNecessities = 0;
  void _calculateZakatNecessities(double
pricePerKg,
double numberOfPeople) {
  setState(() {
    _zakatNecessities = pricePerKg *
(numberOfPeople *
2.8);
    globalHistory.add({
      'type': 'Zakat Fitrah',
      'pricePerKg': pricePerKg,
      'numberOfPeople': numberOfPeople,
      'zakat': _zakatNecessities,
      'date': DateTime.now(),
    });
  });
}
}

```

Kode di atas merupakan bagian dari aplikasi yang mengelola perhitungan zakat fitrah. `ZakatFitrahPage` adalah halaman utama untuk fitur zakat fitrah, yang menggunakan `StatefulWidget` agar dapat mengelola perubahan status secara dinamis. Di dalamnya, `_ZakatFitrahPageState` mengatur logika dan tampilan halaman tersebut, dengan dua kontroler teks untuk menginput kebutuhan dan harga per kilogram. Fungsi `_calculateZakatNecessities` bertugas menghitung jumlah zakat fitrah dengan mengalikan harga per kilogram dengan jumlah orang yang dikalikan 2,8 (nilai zakat fitrah per orang dalam kilogram). Hasil perhitungan disimpan dalam variabel `_zakatNecessities` dan dicatat dalam `globalHistory` untuk referensi di masa depan. Implementasi ini memastikan bahwa perhitungan zakat fitrah dilakukan secara akurat dan hasilnya terdokumentasi dengan baik.

Waris

```

List<dynamic>
jumlahnyaAhliWaris(Map<String, int>
input)
{
  // Membuat Map untuk menyimpan jumlah
ahli waris
Map<String, int> hasil = {};
  // Melakukan iterasi pada input
  // Melakukan iterasi pada pasangan
kunci-nilai
menggunakan entries
for (MapEntry<String, int> entry in
input.entries) {
  String ahliWaris = entry.key;
  int jumlah = entry.value;
  hasil[ahliWaris] = jumlah;
}
}

```

```

}
// Mengembalikan Map hasilMap<String,
String> terhalang = {};
for (var entry in hasil.entries) {
  var awaris1 = entry.key;
  var jumlah = entry.value;
  for (var entry2 in hasil.entries) {
    var key = entry2.key;
    var value = entry2.value;
    if (awaris1 == key) {
      continue;
    } else if (persyaratanTerpenuhi(awaris1,
jumlah,
key) == false) {
      terhalang[awaris1] = '$awaris1
terhalangi oleh
$key';
      break;
    } else if (persyaratanTerpenuhi(awaris1,
jumlah,
key) == true) {
      continue;
    }
  }; // Lakukan sesuatu dengan entry2 jika
kunci tidak
sama dengan awaris1
}
}
hasil.removeWhere((key, value) =>
terhalang.containsKey(key));
return [hasil, terhalang];
}

```

Fungsi ``jumlahnyaAhliWaris`` dirancang untuk menghitung dan menyusun jumlah bagian waris berdasarkan input yang diberikan dalam bentuk peta (map) yang memetakan nama ahli waris ke jumlah bagiannya. Pertama, fungsi ini menginisialisasi peta kosong ``hasil`` untuk menyimpan jumlah ahli waris. Fungsi kemudian mengiterasi setiap pasangan kunci-nilai dari input, menyimpan kunci dan nilai ke dalam peta ``hasil``. Selanjutnya, fungsi melakukan pemeriksaan lebih lanjut untuk setiap ahli waris dalam ``hasil``, memeriksa apakah ada kondisi yang menghalangi hak warisnya menggunakan fungsi ``persyaratanTerpenuhi``. Jika ditemukan adanya penghalang, ahli waris yang terhalangi dicatat dalam peta ``terhalang``. Akhirnya, fungsi menghapus entri dari ``hasil`` yang ada dalam ``terhalang`` dan mengembalikan daftar yang berisi peta ``hasil`` dan ``terhalang``. Fungsi ini memastikan bahwa pembagian waris dilakukan sesuai dengan ketentuan hukum waris Islam, dengan mempertimbangkan berbagai kondisi dan ketentuan yang berlaku.

C. Pengujian

Pengujian dilakukan dalam dua tahap, dimulai dengan menguji berbagai fitur aplikasi untuk memastikan setiap fungsionalitas berfungsi dengan baik. Pengujian ini mencakup pengujian fitur utama aplikasi.

Blackbox Testing

Tabel 4.1 Blackbox Testing Tahap 1

| No | Data Masukkan | Hasil yang Diharapkan | Hasil Pengamatan | Kesimpulan |
|----|---|---|--|------------|
| 1 | Membuka aplikasi | Menampilkan halaman <i>login</i> | Dapat menampilkan <i>login screen</i> | Gagal |
| 2 | Mengisi email dan kata sandi lalu klik tombol masuk (email dan sandi benar dan sudah terdaftar) | Masuk dan menampilkan <i>homepage</i> | Dapat masuk dan menampilkan <i>homepage</i> | Gagal |
| 3 | Mengisi email dan kata sandi lalu klik tombol masuk (email atau sandi salah atau belum terdaftar) | Menampilkan teks " <i>Authentication failed</i> " | Dapat menampilkan teks " <i>Authentication failed</i> " | Gagal |
| 4 | Klik teks " <i>Forgot Password</i> " | Menampilkan menu <i>send email reset password</i> | Dapat mengirimkan <i>email reset password</i> | Gagal |
| 5 | Klik teks "Belum punya akun? Daftar disini" | Menampilkan halaman registrasi | Dapat menampilkan halaman registrasi | Gagal |
| 6 | Mengisi Nama Lengkap, <i>username</i> , email, dan <i>password</i> lalu klik tombol registrasi | Data akan masuk ke <i>database</i> | Data dapat masuk ke <i>database</i> | Gagal |
| 7 | Klik <i>icon calender</i> | Menampilkan halaman <i>reminder</i> | Dapat mengatur tanggal dan waktu notifikasi pengingat pembayaran | Gagal |
| 8 | Klik <i>icon Zakat</i> | Menampilkan menu kalkulator zakat | Dapat menampilkan menu kalkulator zakat | Berhasil |
| 9 | Klik <i>icon Waris</i> | Menampilkan halaman kalkulator waris | Dapat menampilkan halaman kalkulator waris | Berhasil |
| 10 | klik <i>icon 'back'</i> | kembali ke halaman sebelumnya | kembali ke halaman sebelumnya | Berhasil |

| | | | | |
|----|--|--|--|----------|
| 11 | Mengisi 'Total orang yang membayar zakat fitrah' dan 'harga beras per kilo' lalu klik 'Hitung zakat' | menampilkan hasil zakat fitrah yang harus dibayarkan | menampilkan hasil zakat fitrah yang harus dibayarkan | Berhasil |
| 12 | klik <i>icon 'back'</i> | kembali ke halaman sebelumnya | kembali ke halaman sebelumnya | Berhasil |
| 13 | klik lingkaran 'Laki-Laki' | Menampilkan daftar ahli waris dan suami dihilangkan | dapat Menampilkan daftar ahli waris dan suami dihilangkan | Berhasil |
| 14 | klik lingkaran 'Perempuan' | Menampilkan daftar ahli waris dan istri dihilangkan | dapat Menampilkan daftar ahli waris dan istri dihilangkan | Berhasil |
| 15 | klik lama <i>icon 'i'</i> di samping ahli waris | Menampilkan hubungan ahli waris dengan mayat | dapat Menampilkan hubungan ahli waris dengan mayat | Berhasil |
| 16 | klik <i>icon '+'</i> di samping kanan ahli waris | menambahkan ahli waris ke dalam <i>list</i> atau menambahkan jumlahnya | dapat menambahkan ahli waris ke dalam <i>list</i> atau menambahkan jumlahnya | Berhasil |
| 17 | klik <i>icon '-'</i> di samping kanan ahli waris | menghapus ahli waris dari dalam <i>list</i> atau mengurangi jumlahnya | dapat menghapus ahli waris dari dalam <i>list</i> atau mengurangi jumlahnya | Berhasil |
| 18 | klik lama <i>icon 'i'</i> di samping 'Total Harta Mayat' | Menampilkan informasi mengenai harta mayat yang harus dimasukkan | dapat Menampilkan informasi mengenai harta mayat yang harus dimasukkan | Berhasil |
| 19 | klik lama <i>icon 'i'</i> di samping 'Total Hutang Mayat' | Menampilkan informasi mengenai hutang mayat yang harus dimasukkan | dapat Menampilkan informasi mengenai hutang mayat yang harus dimasukkan | Berhasil |

| | | | | |
|----|--|--|--|----------|
| 20 | Mengisi 'Total Harta Mayat' dan 'Total Hutang Mayat' lalu klik 'Hitung Pembayaran Harta Waris' | ke halaman 'Hasil Pembagian' dan menampilkan hasil perhitungan | dapat ke halaman 'Hasil Pembagian' dan menampilkan hasil perhitungan | Berhasil |
| 21 | klik lama <i>icon</i> lingkaran 'i' di samping nominal harta yang dibagikan | Menampilkan informasi mengenai nominal yang dibagikan | dapat Menampilkan informasi mengenai nominal yang dibagikan | Berhasil |

Pada tahap pertama Blackbox Testing, dilakukan serangkaian uji coba yang mendalam terhadap berbagai fitur aplikasi, dengan tujuan untuk memastikan bahwa setiap fungsionalitas yang ada beroperasi sesuai dengan yang diharapkan. Pengujian ini mencakup pemeriksaan menyeluruh terhadap setiap fitur untuk mengidentifikasi dan mengatasi potensi masalah, sehingga aplikasi dapat berfungsi secara optimal di berbagai kondisi penggunaan.

Tabel 4.2 Blackbox Testing Tahap 2

| No | Data Masukkan | Hasil yang Diharapkan | Hasil Pengamatan | Kesimpulan |
|----|---|---|---|------------------|
| 1 | Membuka aplikasi | Membuka aplikasi | Membuka aplikasi | Membuka aplikasi |
| 2 | Mengisi email dan kata sandi lalu klik tombol masuk (email dan sandi benar dan sudah terdaftar) | Masuk dan menampilkan <i>homepage</i> | Dapat masuk dan menampilkan <i>homepage</i> | Berhasil |
| 3 | Mengisi email dan kata sandi lalu klik tombol masuk (email atau sandi salah atau belum terdaftar) | Menampilkan teks " <i>Authentication failed</i> " | Dapat menampilkan teks " <i>Authentication failed</i> " | Berhasil |
| 4 | Klik teks " <i>Forgot Password</i> " | Menampilkan menu <i>send email reset password</i> | Dapat mengirimkan <i>email reset password</i> | Berhasil |

| | | | | |
|----|--|--|---|----------|
| 5 | Klik teks "Belum punya akun? Daftar disini" | Menampilkan halaman registrasi | Dapat menampilkan halaman registrasi | Berhasil |
| 6 | Mengisi Nama Lengkap, <i>username</i> , <i>email</i> , dan <i>password</i> lalu klik tombol registrasi | Data akan masuk ke <i>database</i> | Data dapat masuk ke <i>database</i> | Berhasil |
| 7 | Mengisi Nama Lengkap, <i>username</i> , <i>email</i> , dan <i>password</i> lalu klik tombol registrasi | Data akan masuk ke <i>database</i> | Data dapat masuk ke <i>database</i> | Berhasil |
| 8 | Klik <i>icon</i> Zakat | Menampilkan menu kalkulator zakat | Dapat menampilkan menu kalkulator zakat | Berhasil |
| 9 | Klik <i>icon</i> Waris | Menampilkan halaman kalkulator waris | Dapat menampilkan halaman kalkulator waris | Berhasil |
| 10 | klik <i>icon</i> 'back' | kembali ke halaman sebelumnya | kembali ke halaman sebelumnya | Berhasil |
| 11 | Mengisi 'Total orang yang membayar zakat fitrah' dan 'harga beras per kilo' lalu klik 'Hitung zakat' | menampilkan hasil zakat fitrah yang harus dibayarkan | menampilkan hasil zakat fitrah yang harus dibayarkan | Berhasil |
| 12 | klik <i>icon</i> 'back' | kembali ke halaman sebelumnya | kembali ke halaman sebelumnya | Berhasil |
| 13 | klik lingkaran 'Laki -Laki' | Menampilkan daftar ahli waris dan suami dihilangkan | dapat Menampilkan daftar ahli waris dan suami dihilangkan | Berhasil |
| 14 | klik lingkaran 'Perempuan' | Menampilkan daftar ahli waris dan istri dihilangkan | dapat Menampilkan daftar ahli waris dan istri dihilangkan | Berhasil |
| 15 | klik lama <i>icon</i> lingkaran 'i' di samping ahli waris | Menampilkan hubungan ahli waris dengan mayat | dapat Menampilkan hubungan ahli waris dengan mayat | Berhasil |

| | | | | |
|----|--|---|---|----------|
| 16 | klik icon '+' di samping kanan ahli waris | menambahkan ahli waris ke dalam list atau menambahkan jumlahnya | dapat menambahkan ahli waris ke dalam list atau menambahkan jumlahnya | Berhasil |
| 17 | klik icon '-' di samping kanan ahli waris | menghapus ahli waris dari dalam list atau mengurangi jumlahnya | dapat menghapus ahli waris dari dalam list atau mengurangi jumlahnya | Berhasil |
| 18 | klik lama icon lingkaran 'i' di samping 'Total Harta Mayat' | Menampilkan informasi mengenai harta mayat yang harus dimasukkan | dapat Menampilkan informasi mengenai harta mayat yang harus dimasukkan | Berhasil |
| 19 | klik lama icon lingkaran 'i' di samping 'Total Hutang Mayat' | Menampilkan informasi mengenai hutang mayat yang harus dimasukkan | dapat Menampilkan informasi mengenai hutang mayat yang harus dimasukkan | Berhasil |
| 20 | Mengisi 'Total Harta Mayat' dan 'Total Hutang Mayat' lalu klik 'Hitung Pembayaran Harta Waris' | ke halaman 'Hasil Pembagian' dan menampilkan hasil perhitungan | dapat ke halaman 'Hasil Pembagian' dan menampilkan hasil perhitungan | Berhasil |
| 21 | klik lama icon lingkaran 'i' di samping nominal harta yang dibagikan | Menampilkan informasi mengenai nominal yang dibagikan | dapat Menampilkan informasi mengenai nominal yang dibagikan | Berhasil |

Pada tahap kedua Blackbox Testing, dilakukan pengujian ulang secara menyeluruh setelah perbaikan yang diidentifikasi pada tahap pertama telah diterapkan. Pengujian ini difokuskan untuk memastikan bahwa semua perbaikan telah diterapkan dengan benar dan setiap fitur utama dalam aplikasi berfungsi dengan baik tanpa adanya gangguan. Proses ini melibatkan pengujian intensif terhadap fitur-fitur kunci untuk menjamin bahwa aplikasi memenuhi standar kualitas yang diharapkan dan dapat beroperasi dengan lancar di berbagai kondisi.

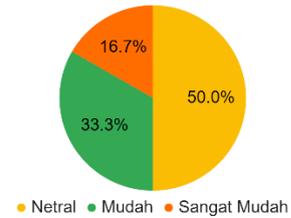
User Acceptance Testing

Skenario: Skenario pengujian ini melibatkan menanyakan beberapa pertanyaan kepada calon pengguna setelah mereka menggunakan aplikasi.

Proses Pengujian: Pengujian ini melibatkan pengguna akhir yang diminta untuk menguji perangkat lunak guna memastikan bahwa aplikasi berfungsi sesuai dengan harapan dan kebutuhan mereka. Pengujian ini bertujuan untuk mengevaluasi tingkat kepuasan pengguna terhadap berbagai aspek fungsionalitas dan antarmuka aplikasi.

Hasil Pengujian:

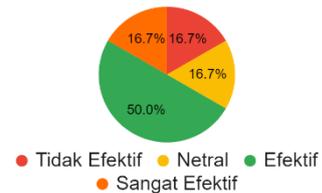
Kemudahan Menemukan dan Menggunakan Ikon Informasi 'i'



Gambar 4.1 Hasil Pertanyaan pertama

Sebagian besar pengguna melaporkan bahwa ikon informasi 'i' cukup mudah ditemukan dan digunakan. Ikon tersebut dinilai cukup intuitif, meskipun masih ada ruang untuk peningkatan agar lebih mudah diakses dan digunakan oleh semua pengguna.

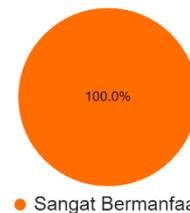
Efektivitas Dialog Box dalam Memahami Konsep Zakat dan Waris



Gambar 4.2 Hasil Pertanyaan Kedua

Hampir semua responden menyatakan bahwa dialog box sangat membantu dalam memahami konsep zakat dan waris dalam Islam. Ini menunjukkan bahwa fitur ini berfungsi dengan baik dalam mendidik pengguna tentang topik yang relevan.

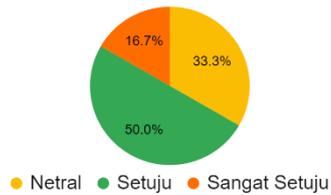
Manfaat Fitur Pengingat Kewajiban Zakat



Gambar 4.3 Hasil Pertanyaan Ketiga

Seluruh responden merasa bahwa fitur pengingat kewajiban zakat sangat bermanfaat. Temuan ini mengindikasikan bahwa fitur tersebut penting dan berhasil dalam membantu pengguna mengingat kewajiban zakat mereka secara efektif.

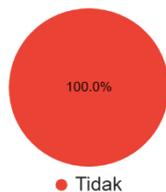
Keterpahaman dan Kemudahan Penggunaan Antarmuka Pengguna



Gambar 4.4 Hasil Pertanyaan Keempat

Sebagian besar responden menyatakan setuju atau sangat setuju bahwa antarmuka pengguna aplikasi ini intuitif dan mudah digunakan. Meskipun demikian, terdapat beberapa saran yang menunjukkan bahwa masih ada ruang untuk perbaikan lebih lanjut dalam desain antarmuka.

Pengalaman Kesulitan Teknis saat Menggunakan Aplikasi



Gambar 4.5 Hasil Pertanyaan Kelima

Seluruh responden melaporkan bahwa mereka tidak mengalami kesulitan teknis saat menggunakan aplikasi ini. Hal ini menunjukkan bahwa aplikasi stabil dan berfungsi dengan baik dari segi teknis, memberikan pengalaman pengguna yang lancar dan tanpa hambatan.

D. Analisa Hasil Pengujian

Tingkat Keberhasilan:

Blackbox Testing: Tahap pertama menunjukkan beberapa fitur tidak berfungsi dengan benar, seperti login dan registrasi. Setelah perbaikan, hampir semua fitur berfungsi dengan baik.

Usability Testing: Aplikasi cukup efisien dan mudah digunakan oleh berbagai kelompok usia, meskipun pengguna yang lebih tua membutuhkan waktu lebih lama untuk menyelesaikan beberapa tugas.

User Acceptance Testing: Aplikasi diterima dengan baik oleh pengguna karena mudah digunakan dan memberikan hasil yang akurat.

Faktor Pendukung:

Blackbox Testing: Pengujian awal yang kurang, serta masalah dalam manajemen database dan autentikasi menjadi penghambat utama.

Usability Testing: Waktu yang diperlukan untuk menyelesaikan beberapa sub-tugas, terutama oleh pengguna yang lebih tua, menjadi hambatan.

User Acceptance Testing: Antarmuka yang perlu disederhanakan dan kebutuhan panduan tambahan menjadi faktor penghambat.

Keterbatasan Solusi:

Blackbox Testing: Perlu revisi kode dan pengujian lebih mendalam untuk beberapa fitur yang masih perlu penyempurnaan.

Usability Testing: Penyederhanaan desain dan penjelasan lebih lanjut dapat meningkatkan efisiensi penggunaan.

User Acceptance Testing: Penyederhanaan antarmuka dan panduan tambahan dapat meningkatkan kenyamanan dan efisiensi penggunaan.

V. KESIMPULAN

Perkembangan teknologi, terutama dalam penerapan kecerdasan buatan (AI), telah memudahkan pengelolaan kewajiban agama seperti zakat dan waris melalui aplikasi kalkulator yang efisien dan akurat. Namun, tantangan signifikan masih ada, terutama dalam menangani kompleksitas hukum waris dan pemahaman pengguna terhadap aturan-aturan Islam. Selain aspek teknis, pentingnya desain antarmuka yang intuitif dan pendidikan pengguna yang komprehensif menjadi kunci keberhasilan aplikasi ini. Keseluruhan pengembangan harus mencakup aspek multidisipliner, dengan kolaborasi antara ahli komputer dan hukum Islam, untuk menghasilkan aplikasi yang tidak hanya fungsional tetapi juga sesuai dengan prinsip-prinsip keagamaan.

Kesimpulan dari rancangan UX tampilan aplikasi ini menekankan pentingnya integrasi antara elemen *User Interface (UI)* dan *User Experience (UX)* dalam menciptakan aplikasi berbasis sistem pakar yang fungsional dan user-friendly. Dalam perancangan aplikasi untuk perhitungan zakat dan waris, prinsip UI/UX diterapkan secara cermat untuk memastikan pengalaman pengguna yang optimal. Desain yang intuitif dan navigasi yang mudah digunakan memungkinkan pengguna dari berbagai latar belakang untuk mengakses fitur-fitur aplikasi dengan efisien. Implementasi fitur-fitur seperti login, registrasi, pop-up harian, dan perhitungan zakat serta waris, didesain dengan mempertimbangkan kenyamanan dan kemudahan pengguna dalam menyelesaikan tugas mereka, memastikan hasil yang akurat dan sesuai dengan ketentuan hukum Islam yang berlaku.

REFERENSI

- [1] R. Setiawan, D. Destiani, and C. Slamet, "Perancangan Sistem Pakar untuk Pembagian Waris Menurut Hukum Islam (Fara'id)," 2012. [Online]. Available: <http://jurnal.sttgarut.ac.id>
- [2] Nadya Febriani Meldi, "LOOKING BACK BERBANTUAN APLIKASI I WARIS PADA MATERI PERBANDINGAN TERINTEGRASI HUKUM WARIS," *Jurnal Alwatzikhoebillah: Kajian Islam, Pendidikan, Ekonomi, Humaniora*, vol. 9, no. 1, pp. 49–69, Jan. 2023, doi: 10.37567/alwatzikhoebillah.v9i1.1540.
- [3] Vinna Lusiana, "HUKUM KEWARISAN DI INDONESIA (Studi Komparatif antara Kompilasi Hukum Islam dengan Kitab Undang-Undang Hukum Perdata)," *Jurnal Alwatzikhoebillah: Kajian Islam, Pendidikan, Ekonomi, Humaniora*, vol. 8, no. 2, pp. 291–306, 2022.
- [4] R. K. Saputri, P. Uin, S. Thaha, and S. Jambi, "Pemahaman Masyarakat Tentang Penerapan Pembagian Harta Warisan Menurut Hukum Waris

- Islam Lisencing,” *Journal of Comprehensive Islamic Studies (JOCIS)*, vol. 2, no. 2, pp. 1–9, 2023, doi: 10.56436/jocis.v2i2.255.
- [5] H. Hidayatulloh, “REALASI ILMU PENGETAHUAN DAN AGAMA,” *Proceedings of The ICECRS*, vol. 1, no. 1, Oct. 2016, doi: 10.21070/picecrs.v1i1.627.
- [6] S. Hakak *et al.*, “Digital Hadith authentication: Recent advances, open challenges, and future directions,” *Transactions on Emerging Telecommunications Technologies*, vol. 33, no. 6, Jun. 2022, doi: 10.1002/ett.3977.
- [7] O. A. Dada and I. T. Sanusi, “The adoption of Software Engineering practices in a Scrum environment,” *African Journal of Science, Technology, Innovation and Development*, vol. 14, no. 6, pp. 1429–1446, Sep. 2022, doi: 10.1080/20421338.2021.1955431.
- [8] T. S. Tajuddin, “COMPARATIVE ANALYSIS OF BUSINESS ZAKAT COMPUTATION OF ISLAMIC FINANCIAL INSTITUTIONS IN MALAYSIA AND SAUDI ARABIA,” *International Journal of Islamic Economics and Finance Research*, vol. 5, no. 1 July, pp. 1–16, Jul. 2022, doi: 10.53840/ijiefer62.
- [9] W. Yihua, F. Meng, M. Farrukh, A. Raza, and I. Alam, “Twelve years of research in *The International Journal of Islamic and Middle Eastern Finance and Management* : a bibliometric analysis,” *International Journal of Islamic and Middle Eastern Finance and Management*, vol. 16, no. 1, pp. 154–174, Jan. 2023, doi: 10.1108/IMEFM-03-2020-0134.
- [10] M. Akbar, A. Akbar, H. S. Yaqoob, A. Hussain, L. Svobodová, and F. Yasmin, “Islamic finance education: Current state and challenges for Pakistan,” *Cogent Economics & Finance*, vol. 11, no. 1, Dec. 2023, doi: 10.1080/23322039.2022.2164665.
- [11] R. Candra Sari, P. L. Rika Fatimah, S. Ilyana, and H. Dwi Hermawan, “Augmented reality (AR)-based sharia financial literacy system (AR-SFLS): a new approach to virtual sharia financial socialization for young learners,” *International Journal of Islamic and Middle Eastern Finance and Management*, vol. 15, no. 1, pp. 48–65, Jan. 2022, doi: 10.1108/IMEFM-11-2019-0484.
- [12] J. Hunt, *Beginner’s Guide to Kotlin Programming*. Cham: Springer International Publishing, 2021. doi: 10.1007/978-3-030-80893-8.
- [13] S. Bose, “A COMPARATIVE STUDY: JAVA VS KOTLIN PROGRAMMING IN ANDROID APPLICATION DEVELOPMENT,” *International Journal of Advanced Research in Computer Science*, vol. 9, no. 3, pp. 41–45, Jun. 2018, doi: 10.26483/ijarcs.v9i3.5978.
- [14] G. B. Hertantyo, W. E. Putra, and M. W. Bayu Fahrizal, “DEVELOPMENT OF AN ANDROID APPLICATION-BASED ACADEMIC INFORMATION SYSTEM AT IMMIGRATION POLYTECHNIC,” *TEMATICS: Technology Management and Informatics Research Journals*, vol. 3, no. 2, pp. 23–53, Dec. 2021, doi: 10.52617/tematics.v3i2.331.