

**MEMBANGUN APLIKASI *MOBILE* PADA ISLAMIC TRUST
FUND (ITF) UIN AR-RANIRY BANDA ACEH**

TUGAS AKHIR

Diajukan oleh :

MUHAMMAD FAIRUZA

200705045

**Mahasiswa Fakultas Sains dan Teknologi
Program Studi Teknologi Informasi**



**FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI AR-RANIRY
BANDA ACEH
2025 M/ 1446 H**

LEMBAR PERSETUJUAN

MEMBANGUN APLIKASI *MOBILE* PADA ISLAMIC TRUST FUND (ITF) UIN AR-RANIRY BANDA ACEH

TUGAS AKHIR

Diajukan kepada Fakultas Sains dan Teknologi UIN Ar-Raniry Banda Aceh
sebagai salah satu persyaratan penulisan Tugas akhir dalam Prodi Teknologi
Informasi

Oleh:


MUHAMMAD FAIRUZA

200705045


**Mahasiswa Fakultas Sains dan Teknologi
Program Studi Teknologi Informasi**

Disetujui untuk dimunaqasyahkan Oleh:

Pembimbing I,



Mulkan Fadli, M.T
NIP: 19881282020121006

Pembimbing II,


Hendri Ahmadian, M.I.M
NIP: 198301042014031002

Mengetahui,

Ketua Program Studi Teknologi Informasi


Malahayati, M.T
NIP: 198301272015032003

LEMBAR PENGESAHAN

MEMBANGUN APLIKASI *MOBILE* PADA ISLAMIC TRUST FUND (ITF) UIN AR-RANIRY BANDA ACEH

TUGAS AKHIR

Telah Diuji Oleh Panitia Ujian Munaqasyah Tugas Akhir
Fakultas Sains dan Teknologi UIN Ar-Raniry Banda Aceh dan Dinyatakan Lulus
Serta Diterima Sebagai Salah Satu Beban Studi Program Sarjana (S-1)
dalam Program Studi Teknologi Informasi

Pada Hari/Tanggal: Selasa, 14 Januari 2025
14 Rajab 1446 H
di Darussalam, Banda Aceh

Panitia Ujian Munaqasyah Tugas Akhir:

Ketua,


Mulkhan Fadhli, M.T
NIP. 198811282020121006


Sekretaris,


Hendri Ahmadian, M.I.M
NIP. 198301042014031002

Penguji I,


Nazaruddin Ahmad, M.T
NIP. 198206052014031002

Penguji II


Muhammad Syamsu Rizal, M.T

Mengetahui:

Dekan Fakultas Sains dan Teknologi
UIN Ar-Raniry Banda Aceh,




Prof. Dr. Ir. M. Dirhamsyah, M.T., I.P.U
NIP. 196210021988111001

LEMBAR PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tanga di bawah ini:


Nama : Muhammad Fairuza
NIM : 200705045
Program Studi : Teknologi Informasi
Fakultas : Sains dan Teknologi
Judul : Membangun Aplikasi *Mobile* Pada Islamic Trust Fund
(ITF) Uin Ar-Raniry Banda Aceh


Dengan ini menyatakan bahwa dalam penulisan tugas akhir ini, saya :

1. Tidak menggunakan ide orang lain tanpa mampu mengembangkan dan mempertanggungjawabkan;
2. Tidak melakukan plagiasi terhadap naskah tugas akhir orang lain;
3. Tidak menggunakan tugas akhir orang lain tanpa menyebutkan sumber asli atau tanpa izin pemilik tugas akhir;
4. Tidak memanipulasi dan memalsukan data;
5. Mengerjakan sendiri tugas akhir ini dan mampu bertanggung jawab atas tugas akhir ini.

Bila dikemudian hari ada tuntutan dari pihak lain atas tugas akhir saya, dan telah melalui pembuktian yang dapat dipertanggungjawabkan dan ternyata memang ditemukan bukti bahwa saya telah melanggar pernyataan ini, maka saya siap dikenai sanksi berdasarkan aturan yang berlaku di Fakultas Sains dan Teknologi UIN Ar-Raniry Banda Aceh.

Demikian pernyataan ini saya buat dengan sesungguhnya dan tanpa paksaan dari pihak manapun.

Banda Aceh, 16 Januari 2025
Yang Menyatakan

(Muhammad Fairuza)



ABSTRAK

Nama : Muhammad Fairuza
NIM : 200705045
Program Studi : Teknologi Informasi
Judul : Membangun Aplikasi *Mobile* Pada Islamic Trust Fund (ITF) UIN Ar-Raniry Banda Aceh
Tanggal Sidang : 14 Januari 2025
Jumlah Halaman : 59 Halaman
Pembimbing I : Mulkan Fadhli, M.T
Pembimbing II : Hendri Ahmadian, M.I.M.
Kata Kunci : Aplikasi *Mobile*, Pengelolaan Zakat, Islamic Trust Fund (ITF), Zakat, Infaq, Sedekah, Wakaf (ZISWAF), *Flutter*.

Islamic Trust Fund (ITF) UIN Ar-Raniry Banda Aceh merupakan lembaga yang berperan dalam mengumpulkan, mengelola, dan menyalurkan dana Zakat, Infaq, Sedekah, dan Wakaf (ZISWAF) guna mendukung kesejahteraan umat. Namun, pengelolaan dana zakat di ITF masih dilakukan secara manual, yang mempengaruhi efisiensi dan keakuratan dalam pencatatan serta penyaluran dana. Penelitian ini bertujuan untuk membangun sebuah aplikasi *mobile* yang dapat memfasilitasi proses pengelolaan dana ZISWAF dengan lebih efisien. Aplikasi ini dirancang menggunakan metode pengembangan perangkat lunak *waterfall*, dengan tahapan analisis kebutuhan, desain sistem, implementasi, dan pengujian sistem. Aplikasi *mobile* ini dibangun menggunakan *Flutter* sebagai framework dan *Firebase* sebagai layanan backend. Hasil penelitian menunjukkan bahwa aplikasi yang dikembangkan mampu meningkatkan efisiensi dalam pengelolaan dana ZISWAF, memperkuat kepercayaan muzakki, serta mendukung tujuan sosial-ekonomi dari zakat itu sendiri di UIN Ar-Raniry Banda Aceh. Dengan adanya aplikasi ini, diharapkan pengelolaan dana ZISWAF di ITF UIN Ar-Raniry dapat dilakukan dengan lebih cepat, mudah, dan sesuai dengan prinsip-prinsip syariah, sehingga kontribusi lembaga ini dalam pemberdayaan ekonomi umat dapat terus meningkat.

ABSTRACT

Name : Muhammad Fairuza
Student : 200705045
Department : Information Technology
Title : *Developing a Mobile Application for the Islamic Trust Fund (ITF) at UIN Ar-Raniry Banda Aceh*
Defense Date : 14 January 2025
Nmber of Pages : 59 Pages
Advisor I : Mulkan Fadhli, M.T
Advisor II : Hendri Ahmadian, M.I.M.
Keywords : *Mobile Application, Zakat Management, Islamic Trust Fund (ITF), Zakat, Infaq, Sadaqah, Waqf (ZISWAF), Flutter.*

The Islamic Trust Fund (ITF) at UIN Ar-Raniry Banda Aceh is an institution responsible for collecting, managing, and distributing Zakat, Infaq, Sadaqah, and Waqf (ZISWAF) funds to support community welfare. However, the management of zakat funds at ITF is still conducted manually, which affects the efficiency and accuracy of record-keeping and fund distribution. This research aims to develop a mobile application to facilitate a more efficient process of ZISWAF fund management. The application is designed using the waterfall software development method, which includes the stages of requirements analysis, system design, implementation, and system testing. This mobile application is built using Flutter as the framework and Firebase as the backend service. The results of the study show that the developed application can improve efficiency in managing ZISWAF funds, strengthen the trust of donors (muzakki), and support the socio-economic goals of zakat itself at UIN Ar-Raniry Banda Aceh. With this application, it is expected that the management of ZISWAF funds at ITF UIN Ar-Raniry can be carried out more quickly, easily, and in accordance with sharia principles, thus enhancing the institution's contribution to the economic empowerment of the community.

KATA PENGANTAR

Puji syukur ke hadirat Allah SWT atas segala limpahan rahmat, hidayah, dan karunia-Nya sehingga penulis dapat menyelesaikan tugas akhir yang berjudul "Membangun Aplikasi *Mobile* pada Islamic Trust Fund (ITF) UIN Ar-Raniry Banda Aceh" ini dengan baik dan tepat waktu. Tugas akhir ini disusun sebagai salah satu syarat untuk menyelesaikan pendidikan Strata 1 (S1) pada Program Studi Teknologi Informasi, Fakultas Sains dan Teknologi, Universitas Islam Negeri (UIN) Ar-Raniry Banda Aceh.

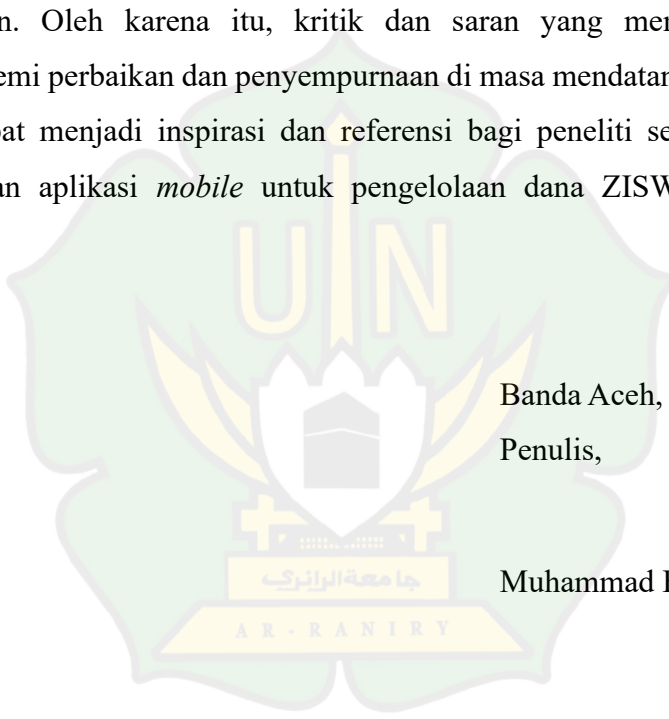
Penulis menyadari bahwa dalam penyusunan tugas akhir ini tidak lepas dari bantuan, dukungan, dan bimbingan dari berbagai pihak. Oleh karena itu, penulis ingin menyampaikan terima kasih yang sebesar-besarnya kepada:

1. Kedua orang tua dan keluarga besar yang telah memberikan dukungan moril, materiil, serta doa yang tiada henti-hentinya, yang menjadi sumber motivasi utama bagi penulis.
2. Bapak Mulkan Fadhli, S.T, M.T., selaku dosen pembimbing utama dan penasihat akademik yang telah memberikan bimbingan, arahan, serta dukungan selama proses penyusunan tugas akhir ini.
3. Bapak Hendri Ahmadian, M.I.M., selaku dosen pembimbing kedua yang telah memberikan panduan, masukan berharga, serta dukungan dalam menyelesaikan tugas akhir ini.
4. Bapak T. Muhammad Ghufuran, S.E., M.E., selaku direktur ITF, yang telah memberikan dukungan serta informasi yang sangat berguna dalam pelaksanaan penelitian ini.
5. Ibu Cut Ida Rahmadiana S.Si., selaku staf Program Studi Teknologi Informasi yang telah membantu dalam berbagai hal *Administratif* dan memberikan dukungan selama masa studi.
6. Seluruh dosen dan staf di Fakultas Sains dan Teknologi UIN Ar-Raniry Banda Aceh yang telah memberikan ilmu pengetahuan dan pengalaman selama masa studi.

7. Teman-teman dan rekan seperjuangan, yang telah memberikan semangat dan motivasi selama proses penyusunan tugas akhir ini.

Penulis berharap, tugas akhir ini dapat memberikan kontribusi nyata dalam pengembangan teknologi informasi, khususnya dalam pengelolaan zakat melalui platform digital di lingkungan akademik UIN Ar-Raniry Banda Aceh. Semoga aplikasi yang dikembangkan dapat bermanfaat bagi masyarakat luas dan menjadi solusi yang efektif dalam pengelolaan dana ZISWAF.

Akhir kata, penulis menyadari bahwa tugas akhir ini masih jauh dari kesempurnaan. Oleh karena itu, kritik dan saran yang membangun sangat diharapkan demi perbaikan dan penyempurnaan di masa mendatang. Semoga tugas akhir ini dapat menjadi inspirasi dan referensi bagi peneliti selanjutnya dalam pengembangan aplikasi *mobile* untuk pengelolaan dana ZISWAF dan bidang lainnya.



Banda Aceh, 16 Januari 2025

Penulis,

Muhammad Fairuza

DAFTAR ISI

LEMBAR PERSETUJUAN	i
LEMBAR PENGESAHAN	ii
LEMBAR PERNYATAAN KEASLIAN TUGAS AKHIR.....	iii
ABSTRAK	iv
ABSTRACT	v
KATA PENGANTAR.....	ivi
DAFTAR ISI.....	viii
DAFTAR TABEL	ix
DAFTAR GAMBAR.....	xii
BAB I PENDAHULUAN.....	1
I.1 Latar Belakang Masalah.....	1
I.2 Rumusan Masalah.....	1
I.3 Tujuan Penelitian.....	2
I.4 Batasan Masalah	2
I.5 Manfaat Penelitian	2
BAB II KAJIAN TEORI	3
II.1 ZISWAF	3
II.2 Universitas Islam Negeri Ar-Raniry	3
II.3 <i>Islamic Trust Fund</i>	4
II.4 <i>Aplikasi Mobile</i>	5
II.5 <i>Waterfall</i>	5
II.6 <i>Figma</i>	6
II.7 <i>Flutter & Dart</i>	7
II.8 <i>Firebase</i>	7
II.9 <i>Visual Video Code</i>	8
II.10 <i>UML</i>	8
II.11 <i>Black Box Testing</i>	8
II.12 Penelitian Terdahulu	9
BAB III METODE PENELITIAN	10
III.1 Tahapan Penelitian	10

III.2 Metode Pengumpulan Data	11
III.2.1 Studi Literatur	11
III.2.2 Observasi	11
III.2.3 Wawancara	11
III.3 Alat Bantu Penelitian.....	12
III.4 Metode Pengembangan Aplikasi	12
III.4.1 Perencanaan	13
III.4.2 Analisis Sistem.....	14
III.4.3 Desain Sistem	16
III.4.4 Implementasi.....	27
III.4.5 Pengujian Sistem.....	28
III.5 Tempat dan Waktu Penelitian	30
BAB IV HASIL PENELITIAN DAN PEMBAHASAN	31
IV.1 Implementasi	31
IV.1.1 Fitur Untuk <i>User</i>	31
IV.1.2 Fitur Untuk <i>Admin</i>	43
IV.2 Pengujian Sistem	54
BAB V KESIMPULAN DAN SARAN	57
V.1 Kesimpulan.....	57
V.2 Saran	57
DAFTAR PUSTAKA.....	58

DAFTAR TABEL

Tabel II.1	Penelitian Terdahulu	9
Tabel III.1	Alat Bantu Penelitian.....	12
Tabel III.2	Hak Akses	13
Tabel III.3	Fitur Aplikasi	14
Tabel III.4	<i>User Story</i>	15
Tabel III.5	Pengujian <i>Black Box</i>	28
Tabel IV.1	Hasil Pengujian <i>Black Box</i>	54



DAFTAR GAMBAR

Gambar III.1	Diagram Alur Penelitian	10
Gambar III.2	Metode Pengembangan Aplikasi	12
Gambar III.3	<i>Use Case</i>	17
Gambar III.4	<i>Class Diagram</i>	18
Gambar III.5	<i>Activity Diagram</i> Pembayaran Zakat.....	19
Gambar III.6	<i>Activity Diagram</i> Infaq	19
Gambar III.7	<i>Activity Diagram</i> Beasiswa	20
Gambar III.8	<i>ERD (Entity Relationship Diagram)</i>	21
Gambar III.9	Relasi Tabel	22
Gambar III.10	<i>Interface Sign In & Daftar Akun</i>	23
Gambar III.11	<i>Interface Menu Lupa Katasandi & Beranda</i>	24
Gambar III.12	<i>Interface Semua Program & Pilihan Berita Penyaluran</i>	25
Gambar III.13	<i>Interface Isi Berita & Pembayaran Zakat</i>	26
Gambar III.14	<i>Interface Pengaturan dan Profil</i>	27
Gambar IV.1	Halaman <i>Login</i>	31
Gambar IV.2	Program Halaman <i>Login</i>	32
Gambar IV.3	Halaman <i>Register</i>	33
Gambar IV.4	Program Halaman <i>Register</i>	34
Gambar IV.5	Halaman Edit Profil.....	35
Gambar IV.6	Program Halaman Edit Profil	35
Gambar IV.7	Halaman Beasiswa.....	37
Gambar IV.8	Program Halaman Beasiswa.....	37
Gambar IV.9	Halaman Pembayaran	38
Gambar IV.10	Program Halaman Pembayaran	39
Gambar IV.11	Halaman <i>Upload</i> Bukti Pembayaran	40
Gambar IV.12	Program Halaman <i>Upload</i> Bukti Pembayaran	41
Gambar IV.13	Halaman Kalkulator Zakat Penghasilan & Emas	42
Gambar IV.14	Program Halaman Kalkulator Zakat Penghasilan & Emas	42
Gambar IV.15	Halaman Tambah Berita	44
Gambar IV.16	Program Halaman Tambah Berita	44
Gambar IV.17	Halaman Tambah Penyaluran	45
Gambar IV.18	Program Halaman Tambah Penyaluran	46
Gambar IV.19	Halaman Tambah Zakat dan Infaq.....	47
Gambar IV.20	Program Halaman Tambah Zakat	47
Gambar IV.21	Program Halaman Infaq	48
Gambar IV.22	Halaman Tambah Beasiswa.....	49
Gambar IV.23	Program Halaman Tambah Beasiswa	50
Gambar IV.24	Halaman Konfirmasi Pembayaran.....	51
Gambar IV.25	Program Halaman Konfirmasi Pembayaran	51
Gambar IV.26	<i>Report</i> Pembayaran <i>User</i>	52
Gambar IV.27	Program <i>Report</i> Pembayaran <i>User</i>	53

BAB I

PENDAHULUAN

I.1 Latar Belakang Masalah

Fenomena tren zakat yang sedang meningkat menunjukkan kesadaran dan antusiasme masyarakat Muslim terhadap kewajiban zakat semakin tinggi. Seiring dengan peningkatan ini, semakin banyak lembaga zakat yang bermunculan, termasuk Lembaga Zakat, Infaq, Shadaqah, dan Wakaf (ZISWAF) seperti *Islamic Trust Fund* (ITF) di UIN Ar-Raniry Banda Aceh. Lembaga-lembaga ini berperan penting dalam mengumpulkan dan menyalurkan zakat kepada yang berhak, mendukung upaya pengentasan kemiskinan, dan meningkatkan kesejahteraan umat.

Namun, di tengah pertumbuhan jumlah lembaga zakat, banyak dari mereka masih menggunakan metode manual dalam menghimpun dan mengelola zakat. Proses manual ini seringkali memakan waktu, tidak efisien, dan rentan terhadap kesalahan serta penyalahgunaan. Tantangan ini menghambat optimalisasi pengumpulan zakat dan mengurangi kepercayaan *muzakki* (pemberi zakat) terhadap lembaga zakat.

Untuk mengatasi masalah ini, solusi yang ditawarkan adalah membangun aplikasi *mobile* untuk menghimpun pembayaran zakat. Aplikasi *mobile* dapat memfasilitasi proses pembayaran zakat secara lebih mudah, cepat, dan aman. Dengan aplikasi ini, *muzakki* dapat melakukan pembayaran zakat kapan saja dan di mana saja, sementara lembaga zakat dapat menampung dana yang terkumpul dengan lebih efisien. Membangun aplikasi *mobile* untuk ITF UIN Ar-Raniry Banda Aceh diharapkan dapat meningkatkan efisiensi pengumpulan zakat, memperkuat kepercayaan *muzakki*, dan mendukung tujuan sosial-ekonomi dari zakat itu sendiri.

I.2 Rumusan Masalah

Dari penjelasan latar belakang di atas, masalah yang muncul adalah penghimpunan zakat yang masih dilakukan secara manual di ITF UIN Ar-Raniry Banda Aceh.

I.3 Tujuan Penelitian

Berdasarkan rumusan masalah, tujuan dari penelitian ini adalah membangun aplikasi *mobile* yang memfasilitasi mengumpulkan dana ZISWAF di ITF UIN Ar-Raniry Banda Aceh.

I.4 Batasan Masalah

Untuk memastikan konsistensi antara judul dan latar belakang yang telah disampaikan, penulis memutuskan untuk membatasi cakupan masalah yang akan dibahas dalam paper ini dengan mengacu pada kutipan berikut.

1. Penelitian ini difokuskan pada penghimpunan dana ZISWAF pada ITF UIN Ar-Raniry Banda Aceh, sehingga solusi yang dihasilkan mungkin tidak sepenuhnya dapat diterapkan pada konteks institusi lain.
2. Pembangunan aplikasi *mobile* hanya dibatasi pada menghimpun dana yang masuk saja.
3. Pembangunan aplikasi *mobile* akan mempertimbangkan ketersediaan sumber daya dan infrastruktur teknologi yang ada, sehingga mungkin terdapat batasan dalam hal fungsionalitas dan fitur yang dapat diimplementasikan.
4. Penelitian ini tidak akan mendalami aspek hukum Islam yang terkait dengan pengelolaan ZISWAF secara mendalam, namun akan lebih berfokus pada aspek teknis pembangunan aplikasi *mobile*.

I.5 Manfaat Penelitian

Berdasarkan latar belakang, rumusan masalah dan tujuan penelitian yang telah penulis uraikan, maka manfaat dari penelitian ini yaitu:

1. Menyediakan solusi dalam bentuk aplikasi *mobile* yang dapat memfasilitasi ITF UIN Ar-Raniry Banda Aceh dalam menghimpun dana yang masuk.
2. Penelitian ini memberikan wawasan tentang penerapan teknologi informasi dalam menghimpun dana ZISWAF, serta dapat menjadi referensi bagi penelitian lanjutan di bidang ini.
3. Penelitian ini membantu peneliti mengembangkan keterampilan analisis sistem dan desain aplikasi, serta memberikan kontribusi terhadap literatur dalam bidang teknologi informasi dan manajemen ZISWAF.

BAB II

KAJIAN TEORI

II.1 ZISWAF

Menurut (Sugita et al., 2020), ZISWAF, yang meliputi Zakat, Infaq, Shodaqah, dan Wakaf, adalah sebuah mekanisme distribusi kekayaan dalam ekonomi Islam. Meskipun hanya zakat yang diwajibkan bagi setiap muslim, ketiga instrumen lainnya merupakan sarana untuk bersedekah sesuai kemampuan individu muslim. Ziswaf memiliki dua makna utama: melaksanakan perintah Tuhan Yang Maha Esa (aspek keagamaan), dan memenuhi tanggung jawab sosial terhadap sesama (aspek kemanusiaan).

Zakat adalah kewajiban bagi umat Islam yang membantu mengurangi kesenjangan ekonomi dengan memberikan sebagian kekayaan kepada yang membutuhkan. Infaq adalah sumbangan sukarela yang diberikan untuk tujuan kebajikan atau kemanusiaan. Shodaqah adalah pemberian sukarela yang dilakukan tanpa batasan tertentu, baik dalam bentuk harta maupun upaya. Sedangkan wakaf adalah pengalihan hak milik harta untuk kepentingan umum atau keperluan sosial, ekonomi, atau keagamaan. Semua instrumen ziswaf ini bertujuan untuk memperbaiki kondisi sosial ekonomi umat serta menciptakan keadilan dan kesejahteraan di masyarakat.

II.2 Universitas Islam Negeri Ar-Raniry

Universitas Islam Negeri (UIN) Ar-Raniry merupakan salah satu perguruan tinggi Islam terkemuka di Indonesia, yang berlokasi di Banda Aceh. Universitas ini didirikan dengan tujuan untuk mengembangkan pendidikan Islam yang berkualitas, serta melahirkan lulusan yang memiliki integritas, keilmuan, dan keterampilan dalam berbagai bidang, termasuk dalam ekonomi dan keuangan Islam. Sebagai institusi pendidikan tinggi yang mengedepankan nilai-nilai Islam, UIN Ar-Raniry juga berperan aktif dalam pemberdayaan masyarakat melalui berbagai program sosial-ekonomi, termasuk pengelolaan zakat, infaq, shadaqah, dan wakaf (ZISWAF).

Dalam konteks peningkatan tren zakat, UIN Ar-Raniry melalui Lembaga ZISWAF *Islamic Trust Fund* (ITF) berkomitmen untuk mengoptimalkan pengelolaan dana zakat guna mendukung kesejahteraan umat. Namun, meskipun ada peningkatan jumlah zakat yang dikumpulkan, pengelolaan dana zakat di ITF UIN Ar-Raniry masih dilakukan secara manual. Proses manual ini mencakup pencatatan penerimaan dan penyaluran zakat, yang seringkali memakan waktu, rentan terhadap kesalahan, dan kurang efisien.

Untuk mengatasi kendala tersebut, solusi yang diusulkan adalah pengembangan aplikasi *Mobile* untuk perhimpunan dan pengelolaan zakat. Aplikasi ini diharapkan dapat mempermudah *muzakki* (pemberi zakat) dalam melakukan pembayaran zakat, serta membantu ITF dalam mengelola dana zakat dengan lebih transparan dan akuntabel. Dengan adanya aplikasi *mobile*, pengelolaan zakat dapat dilakukan secara lebih efisien, akurat, dan sesuai dengan prinsip-prinsip syariah.

Penelitian ini bertujuan untuk merancang dan mengimplementasikan aplikasi *mobile* yang dapat memfasilitasi proses perhimpunan dan pengelolaan zakat di ITF UIN Ar-Raniry Banda Aceh, sehingga dapat meningkatkan kepercayaan masyarakat dan efektivitas program ZISWAF di universitas tersebut.

II.3 *Islamic Trust Fund*

Islamic Trust Fund UIN Ar-Raniry adalah sebuah lembaga yang berfokus pada pengumpulan, pengelolaan, dan distribusi dana Zakat, Infaq, Sadaqah, dan Wakaf (ZISWAF) melalui beragam program. Lembaga ini memiliki nilai-nilai inti seperti Kepercayaan, Kepatuhan Syariah, Keadilan, Inovasi, Kreativitas, Pertumbuhan, Perkembangan, Ibadah, dan Keunggulan.

Dalam konteks Universitas Islam Negeri Ar-Raniry, *Islamic Trust Fund* UIN Ar-Raniry berperan sebagai pengelola yang progresif dalam menyediakan dan mengelola dana zakat, infaq, sadaqah, dan wakaf, dengan tujuan menghasilkan sumber daya manusia yang bertakwa, moderat, cerdas, dan unggul bagi Indonesia.

Dengan semangat "Pertumbuhan dan Pemberdayaan", *Islamic Trust Fund* UIN Ar-Raniry menggalang partisipasi dari pimpinan dan seluruh komunitas

akademik UIN Ar-Raniry Banda Aceh, serta masyarakat di sekitarnya, untuk meningkatkan pembayaran dana ZISWAF dari dosen, pegawai, pelajar, dan warga. Tujuannya adalah agar dana tersebut dapat digunakan secara efektif dan didistribusikan dengan tepat guna untuk meningkatkan kesejahteraan baik di internal (warga kampus) maupun eksternal (masyarakat).

Pada struktur ITF sendiri, ketua dari dewan eksekutifnya dipimpin langsung oleh Wakil Rektor 1 yaitu Prof. Dr. Muhammad Yasir Yusuf, M.A., T. Muhammad Ghufran, S.E., M.E. sebagai direktur, Majidah Nur, S.Sy., M.E. sebagai ketua sekretariat, Fitria Andriani, S.H., M.H. sebagai departemen program & pendanaan dan Mauliza Akbar, M.Ag. sebagai departemen pemasaran.

II.4 Aplikasi *Mobile*

Aplikasi *mobile* adalah perangkat lunak yang dirancang khusus untuk dijalankan pada perangkat seluler seperti ponsel pintar (*smartphone*) atau *tablet*. Aplikasi ini bisa diunduh dan diinstal langsung dari toko aplikasi yang disediakan oleh sistem operasi seluler yang digunakan, seperti *App Store* untuk iOS atau *Google Play Store* untuk Android. Aplikasi *mobile* dapat beragam, mulai dari permainan, media sosial, produktivitas, hingga aplikasi yang membantu dalam berbelanja *online*, pembayaran, dan banyak lagi. Aplikasi *mobile* umumnya membantu pengguna terhubung dengan layanan internet yang biasa diakses melalui komputer, atau memfasilitasi penggunaan aplikasi internet pada perangkat yang dapat dibawa (Firmansyah & Febryansyah, 2021).

II.5 *Waterfall*

Model *waterfall* diperkenalkan pertama kali oleh Winston Royce sekitar tahun 1970. Meskipun dianggap kuno, model ini tetap menjadi salah satu yang paling umum digunakan dalam Rekayasa Perangkat Lunak (SE) saat ini. Model ini mengikuti pendekatan sistematis dan berurutan, dengan setiap tahap menunggu penyelesaian tahap sebelumnya sebelum melanjutkan. Dinamakan "*waterfall*" karena prosesnya bergerak secara linear, dengan setiap tahap mengalir ke tahap berikutnya tanpa mundur ke tahap sebelumnya. Tahapan pengembangan dimulai

dari perencanaan hingga pemeliharaan, dengan setiap tahap memerlukan penyelesaian tahap sebelumnya sebelum melanjutkan (Wahid, 2020).

A. Analysis

Tahap analisis adalah ketika kita mengumpulkan, memahami, dan mendokumentasikan kebutuhan sistem dengan cermat. Ini melibatkan interaksi yang intens dengan pemangku kepentingan untuk memastikan pemahaman yang akurat tentang tujuan sistem.

B. Design

Setelah kebutuhan dipahami, fase desain dimulai. Di sini, kita merencanakan arsitektur sistem dan struktur yang diperlukan. Ini melibatkan pemilihan teknologi yang tepat dan perancangan antarmuka pengguna yang efektif.

C. Implementation

Tahap implementasi melibatkan penerjemahan desain menjadi kode yang dapat dieksekusi. Tim pengembang mulai menulis kode berdasarkan spesifikasi yang telah ditetapkan sebelumnya.

D. Testing

Setelah implementasi selesai, sistem diuji untuk memastikan bahwa itu berfungsi sesuai dengan persyaratan yang telah ditetapkan. Ini melibatkan pengujian fungsional, pengujian kinerja, dan pengujian keamanan.

II.6 Figma

Figma adalah sebuah tool desain yang sering dipakai untuk menciptakan tampilan aplikasi *mobile*, *desktop*, *website*, dan lainnya. *Figma* dapat diakses melalui sistem operasi Windows, Linux, atau Mac dengan koneksi internet. Biasanya, *Figma* digunakan oleh para profesional di bidang UI/UX, desain web, dan bidang terkait lainnya (Muhyidin et al., 2020).

Selain itu, *Figma* juga memungkinkan kolaborasi tim yang efisien dalam proses desain. Fitur kolaborasi real-time memungkinkan beberapa pengguna untuk

bekerja pada proyek desain secara bersamaan, bahkan dari lokasi yang berbeda. Ini memudahkan tim desain untuk berkolaborasi, memberikan umpan balik, dan membuat perubahan secara langsung dalam desain tanpa perlu melakukan banyak iterasi. Hal ini membuat *Figma* menjadi alat yang sangat berguna dalam pengembangan produk digital yang membutuhkan kerja tim yang sinergis dan responsif.

II.7 Flutter & Dart

Flutter merupakan sebuah SDK atau kerangka kerja *open source* yang dibuat oleh Google untuk mengembangkan aplikasi yang dapat dijalankan di sistem operasi Android dan iOS. *Flutter* menggunakan bahasa pemrograman Dart untuk pengkodean. Keunggulan *Flutter* dibandingkan dengan kerangka kerja lainnya terletak pada proses pembangunan aplikasi, di mana semua kode dikompilasi dalam kode native-nya (Android NDK, LLVM, AOT-compiled) tanpa menggunakan interpreter dalam prosesnya, sehingga proses kompilasi menjadi lebih cepat (Krisnada & Tanone, 2020).

Dart adalah bahasa pemrograman yang dikembangkan oleh Google, yang merupakan bahasa umum yang dapat digunakan untuk membangun berbagai platform seperti *web*, *Mobile*, *server*, dan *Internet of Things (IoT)*. Bahasa ini juga merupakan bahasa resmi yang digunakan dalam pengembangan menggunakan *Flutter* (Hanif & Sinambela, 2020).

II.8 Firebase

Firebase, dikembangkan oleh Google, adalah layanan *Backend as a Service (BaaS)* yang membantu pengembang perangkat lunak dalam membangun aplikasi mereka dengan lebih mudah. *Firebase* menyediakan berbagai layanan yang dapat diintegrasikan dalam sistem, seperti otentikasi untuk memungkinkan pengguna *login* dan terhubung langsung dengan *server firebase*, penyimpanan *firebase* yang memungkinkan pengembang untuk mengunduh dan mengunggah berkas, serta *cloud functions firebase* yang memungkinkan kode dijalankan secara otomatis pada *backend* sebagai tanggapan terhadap fitur atau permintaan *HTTPS* yang dipicu. *firebase* juga menyediakan beragam library yang dapat digunakan dengan berbagai

framework lainnya seperti *NodeJS*, *Java*, *JavaScript*, *ReactJS*, dan lain-lain (Panjaitan & Pakpahan, 2021).

II.9 Visual Studio Code

Visual Studio Code (VS Code) adalah sebuah editor kode sumber yang memiliki bobot ringan namun memiliki kekuatan yang besar, yang dapat dijalankan pada desktop dan tersedia untuk sistem operasi *Windows*, *macOS*, dan *Linux*. Editor ini dilengkapi dengan dukungan *built-in* untuk *JavaScript*, *TypeScript*, dan *Node.js*, serta memiliki ekosistem ekstensi yang kaya untuk bahasa pemrograman lainnya seperti *C++*, *C#*, *Java*, *Python*, *PHP*, dan *Go*, serta untuk *runtime* seperti *.NET* dan *Unity* (Romzi & Kurniawan, 2020).

II.10 Unified Modeling Language (UML)

Unified Modeling Language (UML) adalah bahasa pemodelan yang digunakan untuk mendeskripsikan, memvisualisasikan, dan mendokumentasikan sistem perangkat lunak serta proses pengembangannya. Dengan menyediakan notasi grafis yang standar, *UML* memungkinkan pemodelan berbagai aspek sistem, termasuk struktur, perilaku, dan interaksi antar komponen. Terdiri dari berbagai diagram, seperti *class diagram*, *use case diagram*, dan *sequence diagram*, *UML* membantu dalam menganalisis kebutuhan, merancang arsitektur perangkat lunak, serta memudahkan komunikasi antar pengembang dan pemangku kepentingan. Penggunaan *UML* yang konsisten meningkatkan pemahaman sistem dan mendukung dokumentasi yang efektif untuk pemeliharaan dan pengembangan di masa mendatang.

II.11 Black box Testing

Black box testing adalah metode pengujian perangkat lunak yang berfokus pada pengamatan hasil eksekusi menggunakan data uji untuk memeriksa fungsionalitasnya. Jenis pengujian ini menitikberatkan pada evaluasi fungsi sistem tanpa memperhatikan detail internal atau struktur kode. (Zidan dkk., 2022)

II.12 Penelitian Terdahulu

Berikut ini adalah beberapa penelitian sebelumnya yang relevan yang dapat dijadikan referensi:

Tabel II.1 Penelitian Terdahulu

No	Referensi	Judul	Hasil Penelitian
1	(Alfarisi, 2023)	Perancangan Aplikasi Pembayaran Zakat Pada Yayasan Baituzzakah Berbasis Android	Penelitian ini menghasilkan aplikasi pembayaran zakat berbasis Android untuk memmanage zakat yang masuk. Meski membantu pengelolaan zakat, aplikasi ini terbatas pada platform Android dan belum mendukung fitur infaq serta sedekah, sehingga memerlukan pengembangan lanjutan.
2	(Alamsyah & Firman Arif, 2019)	Pendataan Penerima Zakat Kantor Baznas Kabupaten Probolinggo Berbasis Android Gis	Penelitian ini menghasilkan sistem informasi peta bantuan zakat berbasis Android menggunakan Android Studio dan Java. Aplikasi ini mempermudah pengelolaan zakat, namun memiliki kekurangan karena belum tersedia di Google Play Store dan hanya mendukung platform Android.
3	(Imron dkk., 2021)	Perancangan Aplikasi <i>Mobile</i> Zakat dan Infaq Berbasis Android Pada Baznas Kabupaten Tangerang	Penelitian ini mengembangkan aplikasi <i>mobile</i> zakat dan infaq berbasis Android menggunakan HTML 5, PHP, dan CSS, dengan integrasi sistem manajemen pengetahuan untuk pencatatan penerimaan dan distribusi zakat. Meski sejalan dengan penelitian serupa, aplikasi ini terbatas pada platform Android dan belum menyajikan versi siap pakai atau tampilan antarmuka yang representatif.

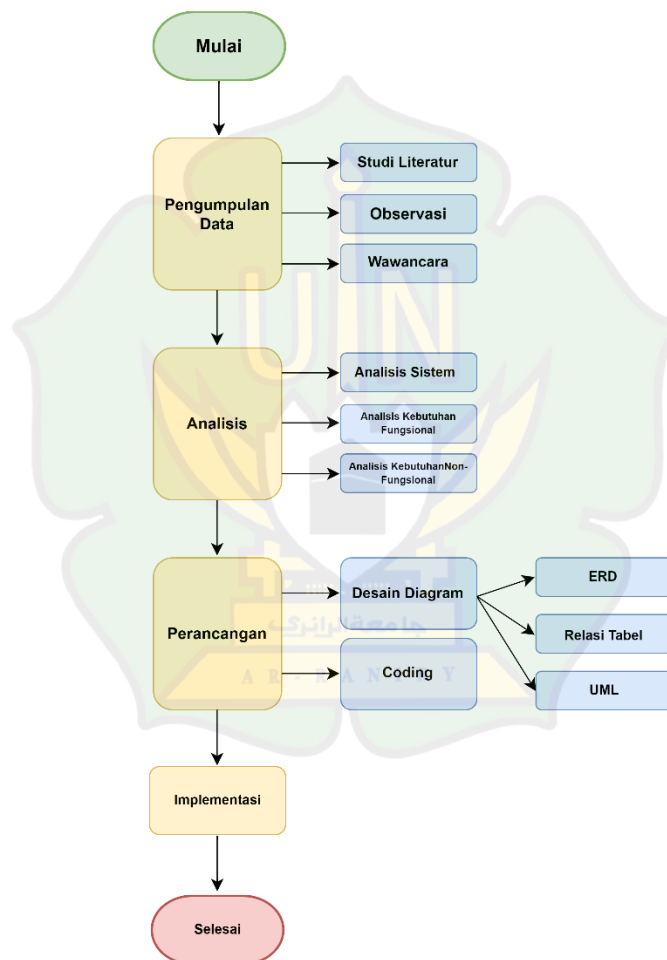
Pada tabel II.1, dapat dilihat penelitian terdahulu yang menjadi referensi pada penelitian ini. Yang menjadi pembeda antara penelitian ini dengan penelitian terdahulu pada tabel II.1 adalah perbedaan dari segi obyek kajian serta teknologi yang digunakan dalam mengembangkan *game*.

BAB III

METODE PENELITIAN

III.1 Tahapan Penelitian

Langkah-langkah penelitian yang akan dilakukan dalam tugas akhir ini dipaparkan pada diagram di bawah ini. Penelitian ini akan mengikuti serangkaian langkah sistematis yang telah direncanakan untuk mencapai tujuan penelitian. Tahapan penelitian dapat dilihat dalam Gambar III.1.



Gambar III.1 Diagram Alur Penelitian

III.2 Metode Pengumpulan Data

Metode pengumpulan data dalam penelitian ini dilakukan untuk mendapatkan informasi yang relevan dan akurat terkait pengelolaan dana ZISWAF di *Islamic Trust Fund* (ITF) UIN Ar-Raniry Banda Aceh. Data dikumpulkan melalui beberapa pendekatan, termasuk studi literatur, observasi, dan wawancara dengan pihak-pihak terkait. Tujuan dari pengumpulan data ini adalah untuk memahami konteks, kebutuhan, serta merancang solusi yang tepat dalam pengembangan aplikasi *mobile* yang mendukung pengelolaan dana ZISWAF secara efisien dan akuntabel.

III.2.1 Studi Literatur

Pada tahapan ini, peneliti menghimpun data atau informasi yang berkaitan dengan ZISWAF dan rancangan aplikasi yang dapat mendukung penelitian ini. Data atau informasi tersebut diperoleh dari berbagai sumber seperti jurnal, buku, penelitian sebelumnya, artikel, dan literatur lainnya.

III.2.2 Observasi

Pada langkah ini, peneliti melakukan observasi terhadap aplikasi ZISWAF yang dihasilkan dari penelitian sebelumnya, serta aplikasi yang tersedia di toko aplikasi saat ini. Tujuannya adalah untuk mengevaluasi apakah aplikasi-aplikasi tersebut menerapkan atau

III.2.3 Wawancara

Wawancara merupakan langkah kunci dalam proses pengumpulan data. Melalui wawancara, pewawancara akan berinteraksi dengan berbagai stakeholder terkait pengelolaan dana amanah ITF, seperti Administrator dana amanah, pengurus ITF. Tujuan utama dari wawancara adalah untuk mendapatkan pemahaman yang lebih mendalam tentang proses pengelolaan dana amanah, kendala yang dihadapi, harapan dan kebutuhan stakeholder terkait, serta evaluasi terhadap solusi yang diajukan.

III.3 Alat Bantu Penelitian

Dalam pelaksanaan penelitian ini, digunakan berbagai alat bantu, termasuk perangkat keras dan perangkat lunak. Perangkat keras utama yang dipergunakan adalah sebuah laptop. Sementara itu, perangkat lunak yang digunakan meliputi sistem operasi Windows 11 beserta berbagai perangkat lunak pendukung yang relevan. Rincian lengkap mengenai alat bantu yang digunakan dapat ditemukan pada tabel di bawah ini.

Tabel III.1 Alat Bantu Penelitian

No	Hardware	Komponen	
1	Laptop Lenovo	Processor	Intel(R) Core(TM) i5-1035G1 CPU @ 1.00GHz 1.19 GHz
2	Laptop Lenovo	Ram	8.00 GB
3	Laptop Lenovo	Storage	512 GB
4	Laptop Lenovo	VGA	NVIDIA GeForce MX330

No	Hardware	Software	Version
1	Laptop Lenovo	<i>Visual Studio Code</i>	v1.89
2	Laptop Lenovo	<i>Flutter</i>	v3.88.0
3	Laptop Lenovo	<i>Dart</i>	v3.88.1
4	Laptop Lenovo	<i>Figma</i>	v116..18.6

III.4 Metode Pengembangan Aplikasi

Dalam metode *waterfall*, terdapat serangkaian tahapan yang dijelaskan secara detail dalam BAB II TINJAUAN PUSTAKA. Pada metode *waterfall* memiliki beberapa tahapan yang harus dilalui sesuai berurutan. Tahapan itu terdiri atas 6 tahap yaitu: Analysis, Design, Implementation, Testing. Berikut juga bisa dilihat berupa rincian tahapan-tahapan pengembangan aplikasi dengan metode *waterfall*.



Gambar III.2 Metode Pengembangan Aplikasi

III.4.1 Perencanaan

Perencanaan sistem melibatkan serangkaian langkah strategis yang mencakup perancangan, pengorganisasian, dan pemodelan sistem. Fokus utama dari proses ini adalah untuk menjelaskan bagaimana sistem akan beroperasi, berkembang, dan berfungsi. Pada bagian ini, terdapat tiga poin penting yang akan diperinci oleh peneliti, diantaranya:

1. Menu Utama

- Beranda
 - Info Berita
 - Highlight Program
 - Highlight Penyaluran
- Program
 - Zakat Profesi
 - Donasi
 - Beasiswa
- Penyaluran
 - Riwayat berita zakat yang disalurkan
- Profil

Akses profil pengguna

2. Akses User & Admin

Tabel III.2 Hak Akses

Menu		Hak Akses	
		User	Admin
No.	Beranda		
1	Info Berita		✓
2	Highlight Program		✓
3	Highlight Penyaluran		✓
No.	Program		
1	Zakat Profesi	✓	✓
2	Donasi	✓	✓
3	Beasiswa	✓	✓
No.	Penyaluran		
1	Riwayat berita zakat yang disalurkan		✓

No.	Profil		
1	Akses profil pengguna	✓	✓

3. Kebutuhan Penggunaan Software

Kebutuhan dalam perangkat lunak mengacu pada situasi di mana satu program atau bagian perangkat lunak memerlukan program atau bagian lainnya untuk berfungsi dengan efektif. Ini dapat dibagi menjadi beberapa sebab:

- Mebutuhkan perangkat *mobile* untuk mengunduh dan meinstal aplikasi.
- Mebutuhkan akses internet untuk menggunakan aplikasi.
- Mebutuhkan akun untuk mengakses aplikasi.

III.4.2 Analisis Sistem

1. Kebutuhan Fungsional

Kebutuhan fungsional untuk pengembangan aplikasi meliputi beberapa aspek penting. Ini termasuk kemampuan autentikasi pengguna, manajemen profil, program, penyaluran dan beranda. Kebutuhan ini dirancang untuk memastikan bahwa aplikasi memiliki fitur yang diperlukan untuk memenuhi kebutuhan pengguna dan tujuan proyek secara keseluruhan. Penjelasan fitur-fitur yang akan dikembangkan terdapat dalam Tabel III.4, di mana nantinya akan diuraikan cerita pengguna (*user stories*) untuk setiap fitur tersebut.

Tabel III.3 Fitur Aplikasi

No.	Fitur	Deskripsi
1	<i>Authentication</i>	Fitur ini akan mengurus segala kebutuhan proses otentikasi pengguna seperti masuk, keluar, dan mendaftar.
2	<i>User Profile</i>	Fitur ini akan mengelola segala kebutuhan terkait menampilkan data pengguna yang sudah terdaftar, serta memenuhi kebutuhan untuk mengedit profil pengguna.
3	Program	Fitur ini akan menangani terkait program-program utama yang menjadi nilai jual atau fungsi dari aplikasi ini.
4	Penyaluran	Fitur ini akan menangani kebutuhan terkait informasi dari progres penyaluran zakat sampai ke penerima.
5	Beranda	Fitur ini untuk menampilkan hal utama pada fitur-fitur lainnya.

Berikut *user stories* dari kebutuhan fitur yang telah dikategorikan dapat dilihat pada Tabel III.5

Tabel III.4 *User Story*

Aktor	Judul <i>User Story</i>	Deskripsi
<i>User</i>	<i>Sign In</i>	Sebagai seorang muzakki, saya ingin <i>Login</i> ke aplikasi agar saya dapat mengakses akun saya.
	<i>User Profile Detail</i>	Sebagai seorang muzakki, saya ingin mengubah profil saya agar informasi saya selalu terkini.
	<i>Edit Photo Profile</i>	Sebagai pengguna saya bisa bebas mengganti foto saya.
	Zakat	Sebagai seorang muzakki, saya ingin melakukan pembayaran zakat secara online agar lebih mudah dan cepat.
	Beasiswa	Sebagai pengguna, saya juga ikut program beasiswa selama saya memenuhi syarat.
	Donasi	Sebagai pengguna, saya juga ikut program donasi yang sudah disediakan alurnya.
	Information	Sebagai seorang muzakki, saya ingin melacak donasi saya agar saya dapat melihat riwayat pembayaran zakat saya.
	<i>Sign In</i>	Sebagai seorang muzakki, saya ingin <i>Login</i> ke aplikasi agar saya dapat mengakses akun saya.
<i>Admin</i>	<i>User Profile Detail</i>	Sebagai <i>Admin</i> , saya ingin mengelola data pengguna agar saya dapat mengupdate atau menghapus data jika diperlukan.
	Zakat	Sebagai <i>Admin</i> , saya ingin memverifikasi pembayaran zakat agar transaksi bisa dikonfirmasi.
	Beasiswa	Sebagai <i>Admin</i> , saya ingin memverifikasi aktivitas program beasiswa agar transaksi bisa dikonfirmasi.
	Infaq	Sebagai <i>Admin</i> , saya ingin memverifikasi pembayaran donasi agar transaksi bisa dikonfirmasi.
	<i>Information</i>	Sebagai <i>Admin</i> , saya ingin memmanage kategori zakat dan infaq agar informasi yang disediakan selalu up-to-date.

2. Kebutuhan Non-Fungsional

Kebutuhan non-fungsional merujuk pada batasan-batasan terkait dengan fitur atau fungsi yang disediakan oleh aplikasi, termasuk aspek software dan hardware yang digunakan dalam pengembangannya (Firdaus, 2023).

Dalam penelitian ini, berikut adalah perangkat lunak yang digunakan dalam pengembangan aplikasi *mobile* ITF:

- a. Visual Studio Code
- b. *Flutter & Dart*
- c. *Firebase*
- d. *Figma*

Dalam pengembangan aplikasi *mobile* ITF pada penelitian ini, berikut adalah beberapa perangkat keras yang digunakan:

- a. Lenovo Ideapad 3i (2020)
- b. Windows 11
- c. RAM 8 GB
- d. Minimal *Disk Space* 5 GB

III.4.3 Desain Sistem

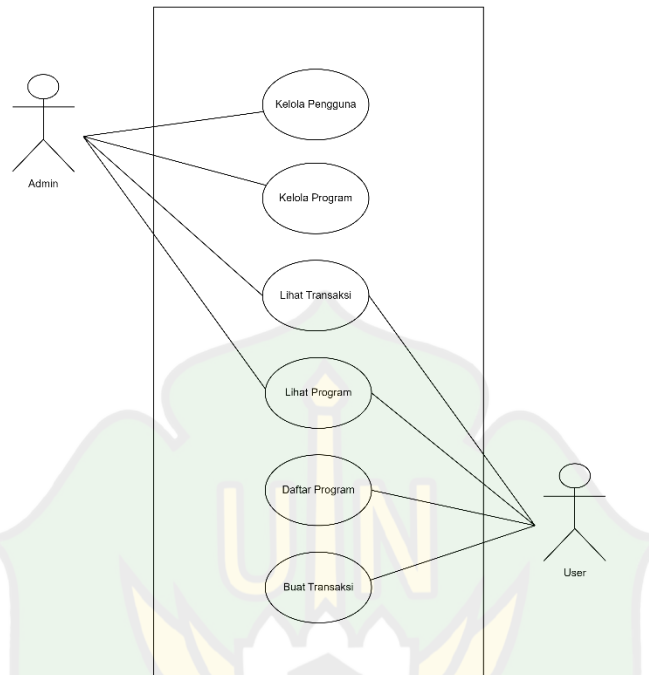
Setelah menyelesaikan analisis kebutuhan, langkah selanjutnya adalah tahap perancangan sistem. Pada tahap ini, terlibat berbagai aspek seperti desain arsitektur sistem, perancangan database dan perancangan antarmuka pengguna. Perancangan sistem pada aplikasi ITF *mobile* akan mencakup elemen-elemen penting seperti pengembangan struktur database yang sesuai, identifikasi entitas yang terlibat dalam sistem, serta penentuan tata letak dan navigasi dalam antarmuka pengguna yang disesuaikan dengan kebutuhan dan tujuan sistem.

Dalam perancangan aplikasi ITF *mobile*, terdapat dua komponen utama yang harus dipertimbangkan, yaitu perancangan sistem secara keseluruhan dan perancangan antarmuka pengguna.

1. Use Case

Use case diagram ini menggambarkan bagaimana Admin dan Pengguna berinteraksi dengan sistem. Admin memiliki peran untuk Kelola Pengguna, Kelola Program, dan Lihat Transaksi, yang berarti mereka bertanggung jawab atas manajemen data pengguna, program, serta memantau transaksi yang terjadi. Sementara itu, Pengguna dapat Lihat Program, Daftar Program, dan Buat Transaksi,

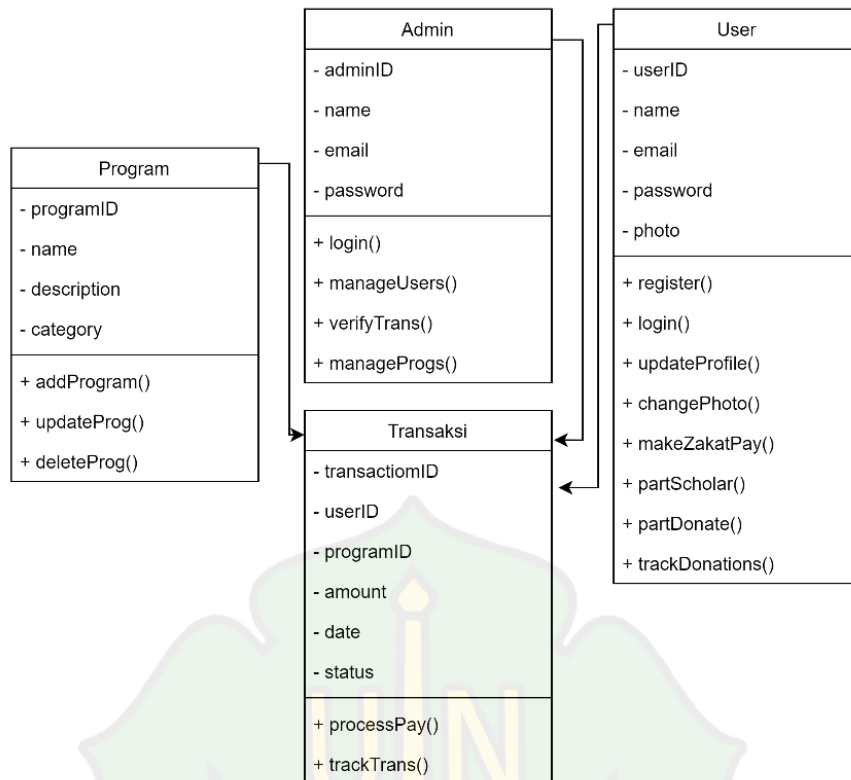
menunjukkan bahwa mereka dapat melihat daftar program, mendaftar ke program tertentu, serta melakukan transaksi terkait program tersebut. Diagram ini memberikan gambaran sederhana tentang fungsi utama yang dapat dilakukan oleh kedua aktor dalam sistem.



Gambar III.3 Use Case

2. Class Diagram

Diagram kelas adalah representasi visual dari struktur statis sistem perangkat lunak yang menggambarkan kelas, atribut, dan hubungan antara kelas-kelas tersebut. Diagram ini memberikan gambaran tentang entitas-entitas yang ada dalam sistem, hubungan antara entitas tersebut, serta properti dan metode yang dimiliki oleh setiap kelas. Dengan menggunakan diagram kelas, pengembang perangkat lunak dapat memvisualisasikan dan merencanakan struktur perangkat lunak sebelum implementasi dimulai.



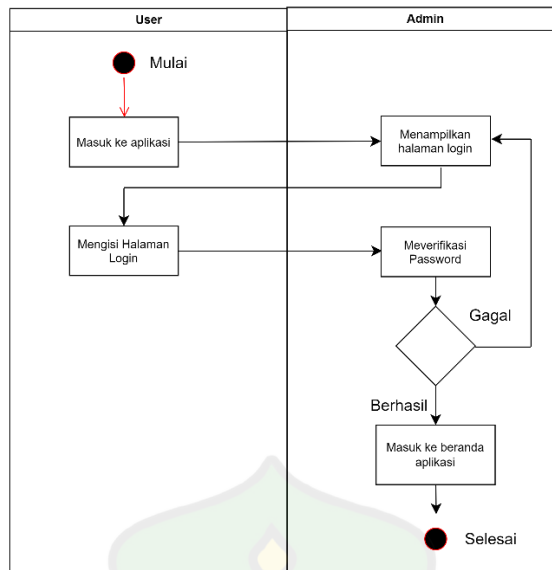
Gambar III.4 Class Diagram

3. Activity Diagram

Dalam rancangan aplikasi, *use case Admin* bertujuan untuk memberikan akses dan mengelola fitur Administratif, termasuk pengelolaan pengguna, konten, dan tindakan Administratif lainnya. Sementara itu, *use case user* menggambarkan interaksi pengguna dengan fitur dan fungsionalitas dalam sistem, seperti membuat profil, melakukan pencarian, atau menggunakan layanan aplikasi sesuai kebutuhan.

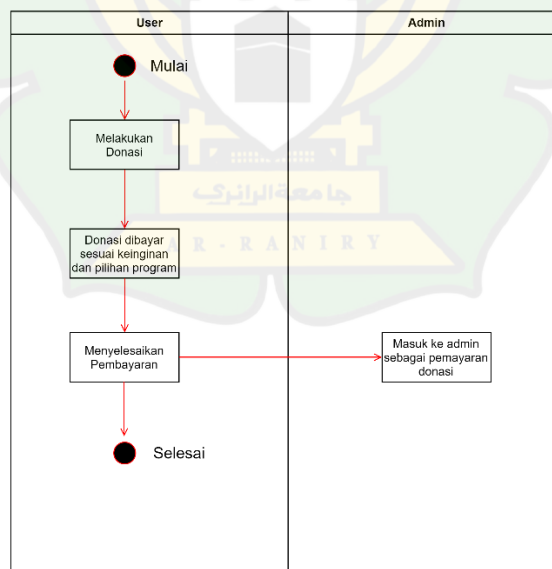
Namun, pada diagram kali ini peneliti membuat beberapa diagram *use case user* dan *Admin*, bisa dilihat pada gambar berikut:

- *Activity Diagram Login*



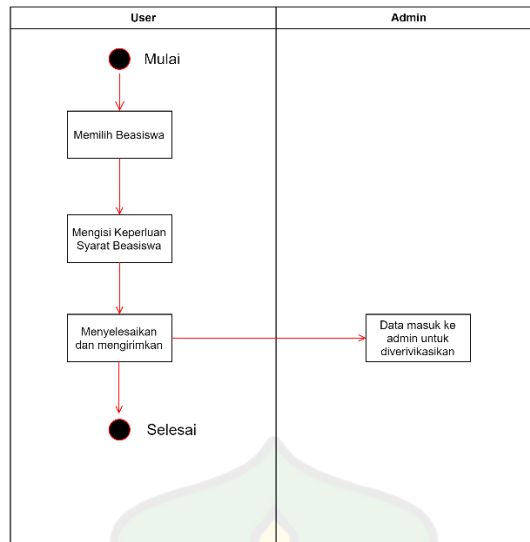
Gambar III.5 Activity Diagram Pembayaran Zakat

- *Activity Diagram Infaq*



Gambar III.6 Activity Diagram Infaq

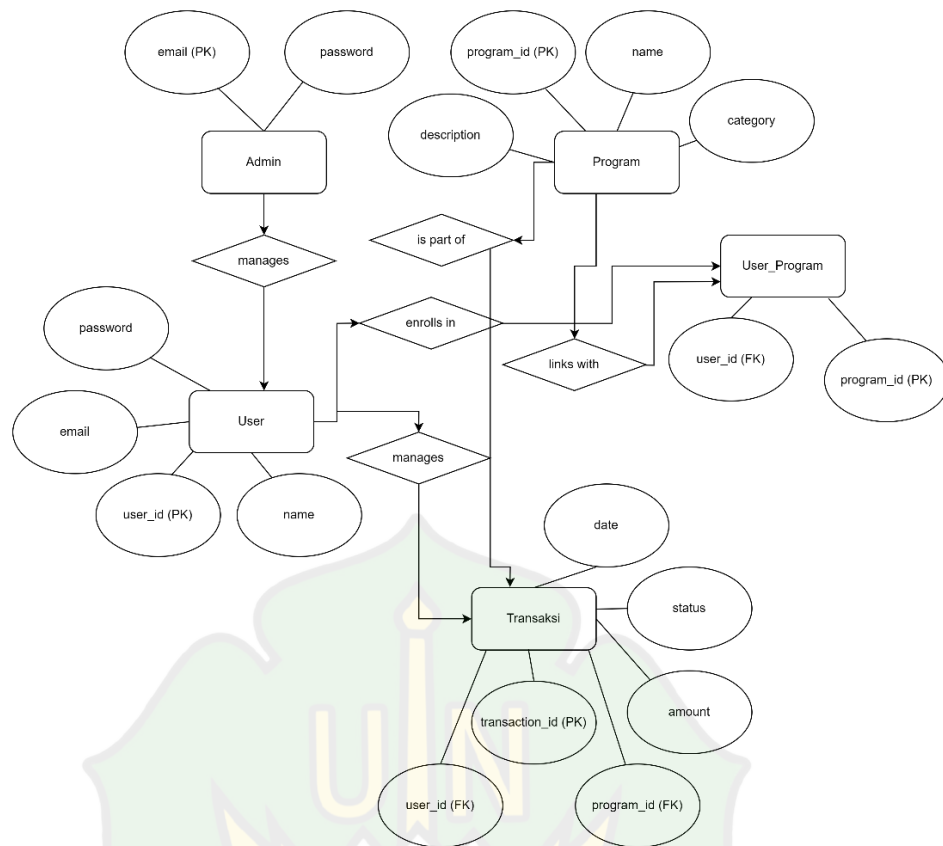
- *Activity Diagram* Beasiswa



Gambar III.7 *Activity Diagram* Beasiswa

4. Entity Relationship Diagram (ERD)

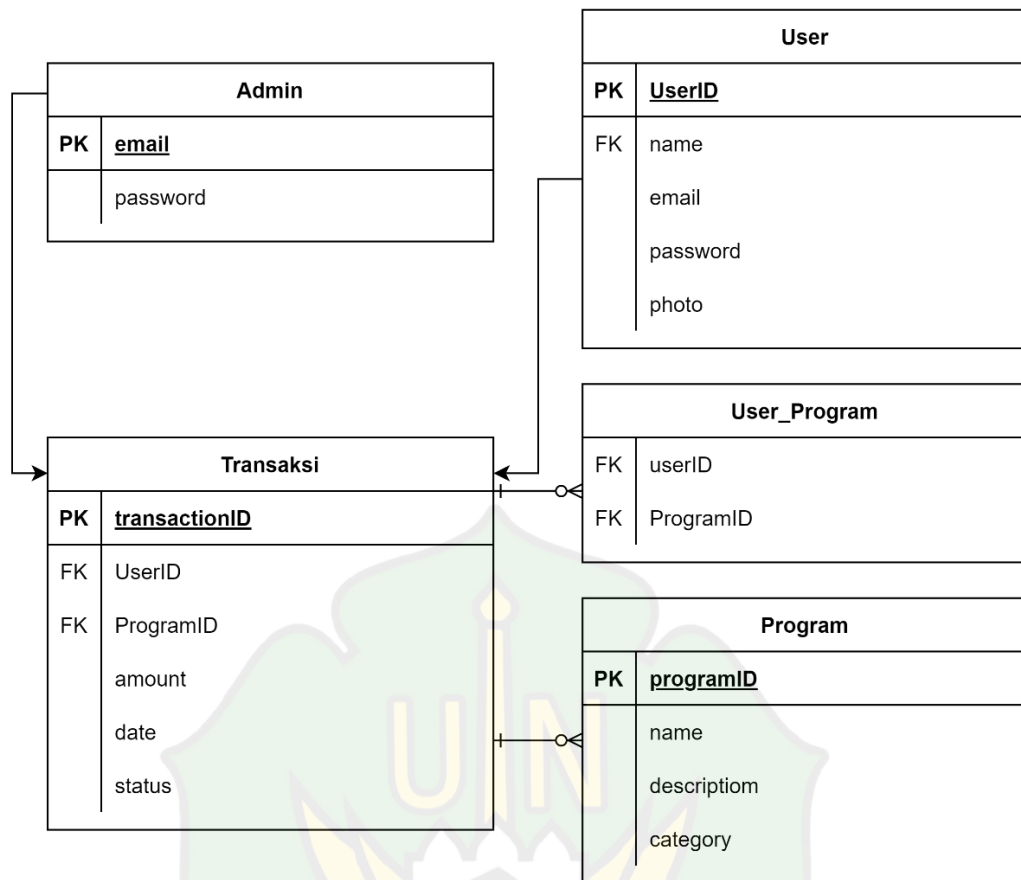
Diagram pada gambar III.8 memberikan representasi yang lebih jelas mengenai relasi antara tabel-tabel dalam sistem aplikasi ITF *mobile*, termasuk bagaimana *Primary Keys* (PK) dan *Foreign Keys* (FK) digunakan untuk menghubungkan tabel-tabel tersebut.



Gambar III.8 Entity Relationship Diagram (ERD)

5. Relasi Tabel

Diagram Relasi tabel pada gambar menunjukkan bahwa *Admin* mengelola data *User*, di mana setiap *User* dapat melakukan banyak *Transaksi*. *User* juga dapat mengikuti banyak *Program* melalui tabel perantara *User_Program*, yang menghubungkan hubungan banyak ke banyak antara *User* dan *Program*. Selain itu, setiap *Transaksi* berhubungan dengan satu *Program* tertentu. Struktur ini dirancang untuk mengatur hubungan antara pengguna, administrator, transaksi, dan program secara terorganisir.



Gambar III.9 Relasi Tabel

6. Rancangan *User Interface*

Diagram *User Interface* akan dirancang dengan memperhatikan kebutuhan dan preferensi pengguna serta mengikuti prinsip-prinsip desain yang baik. Ini meliputi penyusunan tata letak yang jelas, navigasi yang mudah dipahami, penggunaan elemen visual yang konsisten, dan penempatan konten yang strategis.

A. Halaman *Sign In* & Daftar Akun

ISLAMIC TRUST FUND
UIN AR-RANIRY

Selamat Datang Masuk ke Akunmu

Email
Masukkan Email

Password
Masukkan Katasandi

[Lupa Katasandi?](#)

Masuk

Belum punya akun? [Mendaftar](#)

ISLAMIC TRUST FUND
UIN AR-RANIRY

Ayo Mulai! Buat Akunmu

Nama
Masukkan Nama

Email
Masukkan Email

Katasandi
Masukkan Katasandi

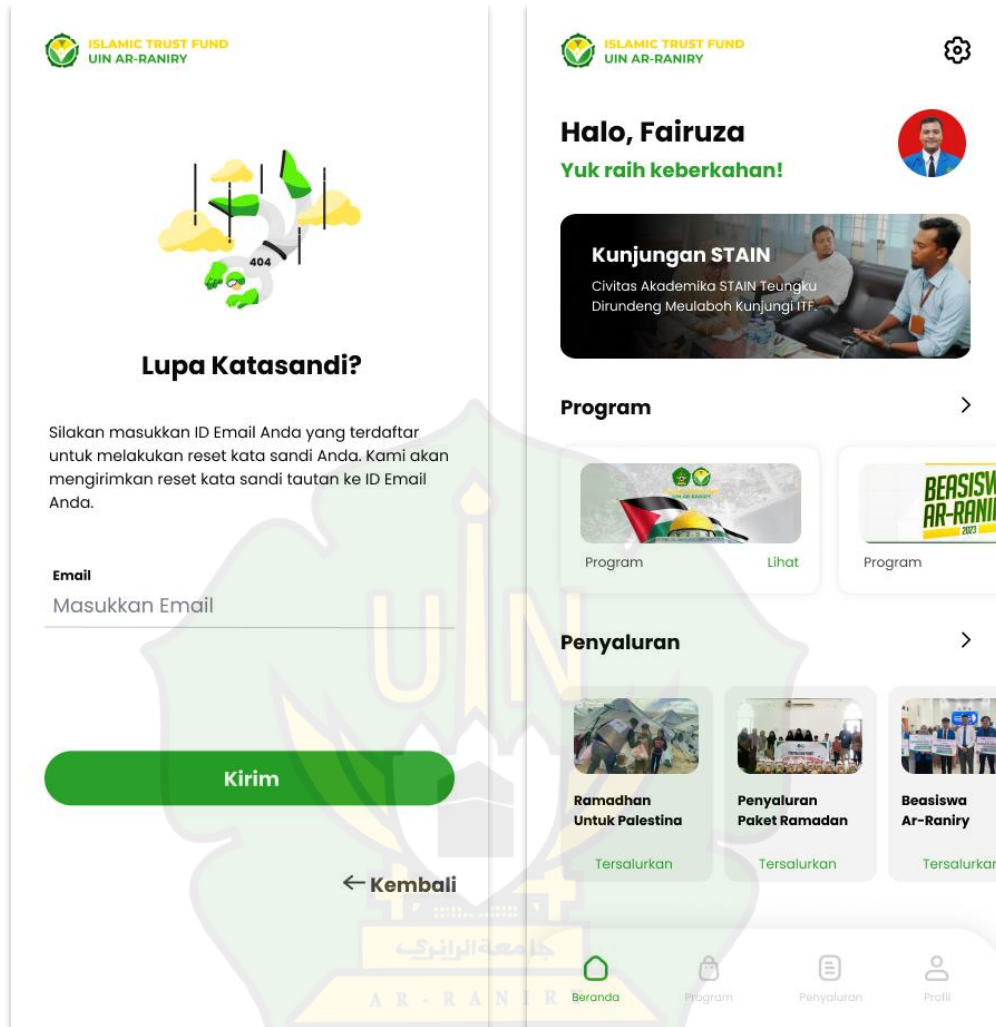
Katasandi
Konfirmasi Katasandi

Mendaftar

Sudah punya akun? [Masuk](#)

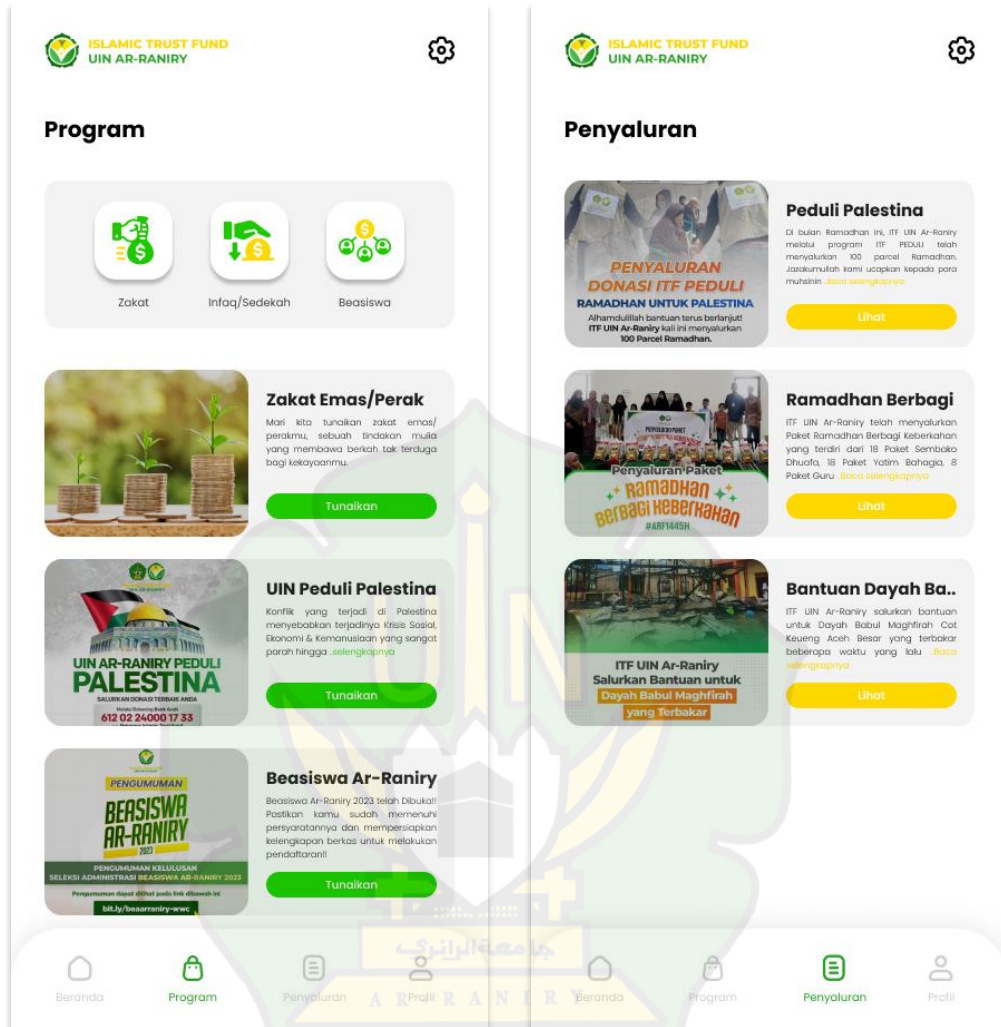
Gambar III.10 *Interface Sign In* & Daftar Akun

B. Halaman *Interface* Menu Lupa Katasandi & Beranda



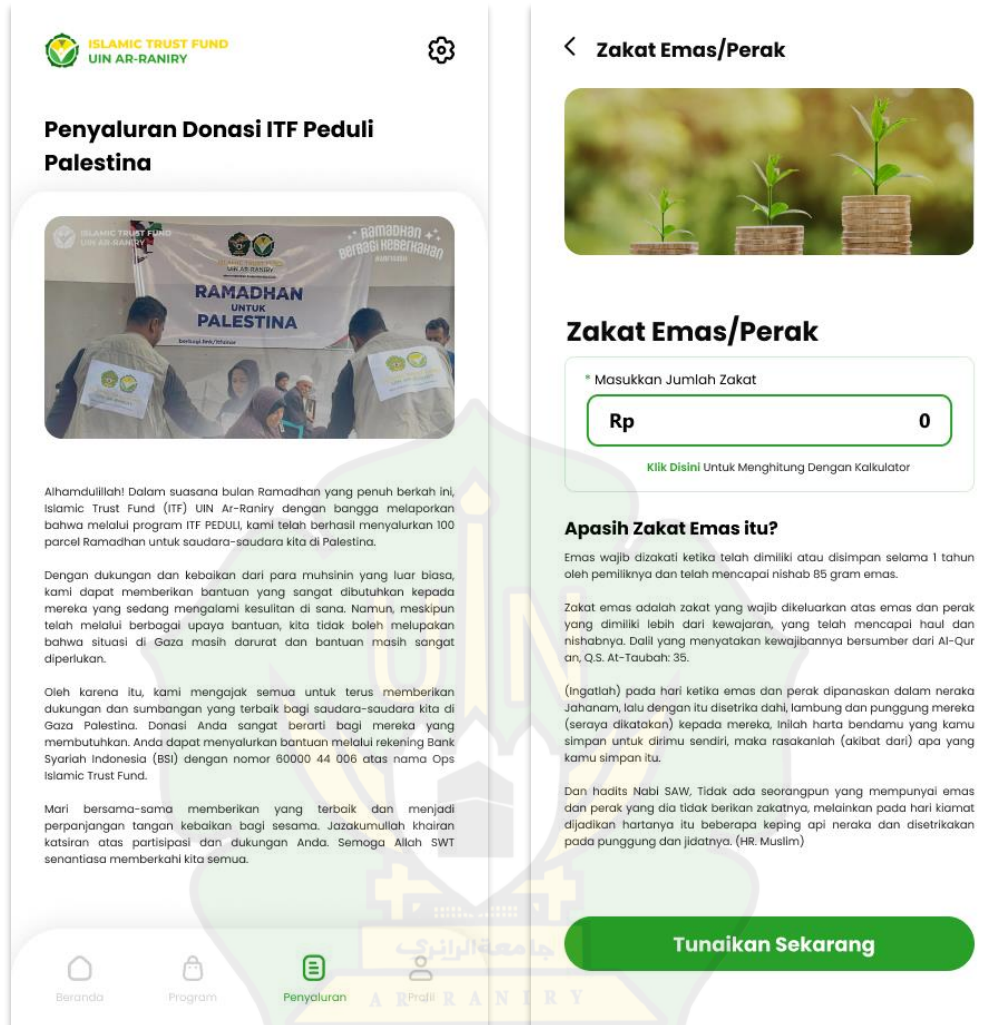
Gambar III.11 *Interface* Menu Lupa Katasandi & Beranda

C. Halaman *Interface* Semua Program & Pilihan Berita Penyaluran



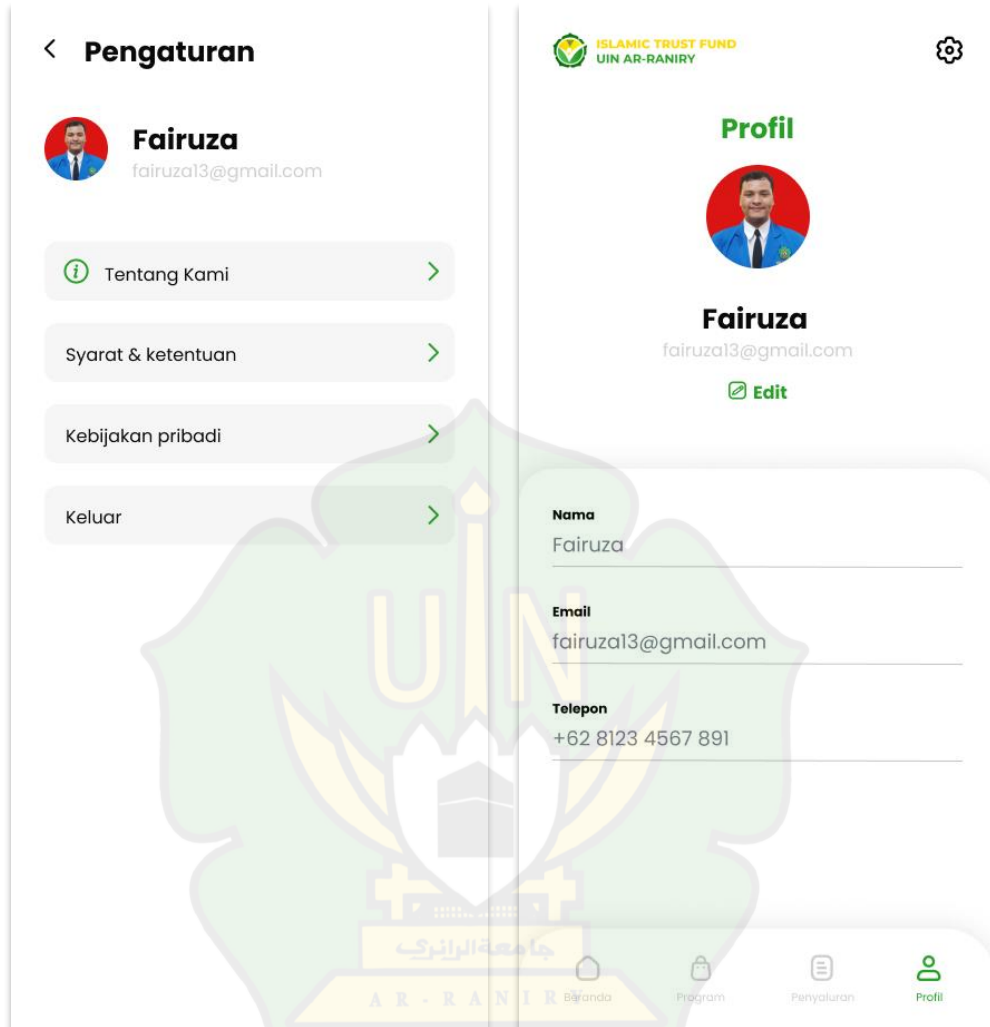
Gambar III.12 *Interface* Semua Program & Pilihan Berita Penyaluran

D. Halaman *Interface* Isi Berita Penyaluran



Gambar III.13 *Interface* Isi Berita Penyaluran & Pembayaran Zakat

E. Halaman *Interface* Pengaturan & Profil



Gambar III.14 *Interface* Pengaturan & Profil

III.4.4 Implementasi

Pada tahap ini, dilakukan implementasi sistem yang sebelumnya telah direncanakan dan dibuat, serta melalui serangkaian pengujian yang diperlukan. Proses instalasi dan konfigurasi sistem akan dilakukan pada server dan database yang telah dipersiapkan sebelumnya. Setelah sistem berhasil terpasang, dilakukan pengujian fungsionalitas untuk memastikan bahwa sistem dapat beroperasi sesuai dengan persyaratan yang telah ditetapkan selama analisis kebutuhan. Jika terjadi

kesalahan atau bug, dilakukan langkah perbaikan dan pengujian ulang sebelum sistem dianggap siap untuk diimplementasikan secara penuh. Hal ini bertujuan untuk memastikan bahwa sistem berjalan dengan lancar tanpa masalah ketika digunakan secara aktif.

III.4.5 Pengujian Sistem

Setelah pengembangan aplikasi selesai, tahap penting selanjutnya adalah pengujian aplikasi menggunakan metode *black box*. Metode ini merupakan teknik pengujian perangkat lunak di mana penguji mengevaluasi fungsi aplikasi tanpa melihat kode internalnya. Fokus utama pengujian *black box* adalah pada input yang diberikan ke sistem dan output yang dihasilkan, untuk memastikan aplikasi berfungsi sesuai dengan spesifikasi yang telah ditentukan. Langkah pertama dalam pengujian ini adalah mengidentifikasi semua fitur utama aplikasi, seperti pendaftaran pengguna dan lain sebagainya.

Tabel III.5 Pengujian *Black Box*

No	Aktor	Skenario Pengujian	Uji Kasus	Hasil Yang Diharapkan
1	<i>User</i>	Pengujian fungsi <i>Registrer</i> .	Mendaftarkan nama, email, katasandi.	Akun berhasil terbuat.
2	<i>User</i>	Pengujian fungsi <i>Login</i> .	Mengisi email dan katasandi yang benar.	Sistem melakukan <i>Login</i> , <i>user</i> masuk ke halaman utama.
3	<i>User</i>	Pengujian fungsi <i>Login</i> .	Mengisi email dan katasandi yang salah.	Sistem akan mengirimkan pesan kesalahan.
4	<i>User</i>	Pengujian pembayaran zakat & infaq.	Melakukan transaksi pembayaran dan meng <i>Upload</i> bukti pembayaran.	Pembayaran bisa selesai terlaksana dan bisa meng <i>Upload</i> bukti pembayaran sampai terverifikasi dari <i>Admin</i> .
5	<i>User</i>	Pengujian riwayat pembayaran	Melihat status dari transaksi yang dilakukan <i>user</i> .	Informasi transaksi bisa terlihat, <i>user</i> dapat melihat status pembayaran dan bukti transfer yang pernah di <i>Upload</i> dan juga bisa meng <i>Upload</i> bukti pembayaran jika

No	Aktor	Skenario Pengujian	Uji Kasus	Hasil Yang Diharapkan
				sebelumnya belum diUpload.
6	User	Pengujian kalkulator zakat.	Mengisi sejumlah bilangan pada kalkulator untuk menghitung jumlah zakat yang dikeluarkan.	Kalkulator berhasil menghitung jumlah zakat yang perlu dikeluarkan oleh user.
7	User	Pengujian Beasiswa	Memilih beasiswa yang diinginkan.	Beasiswa berhasil dipilih dan user diarahkan ke google form untuk mengisi formulir beasiswa yang dipilih.
8	Admin	Pengujian input Berita.	Menambahkan Berita.	Berita berhasil ditambahkan dan bisa tampil dihalaman utama.
9	Admin	Pengujian input Penyaluran.	Menambahkan Penyaluran.	Penyaluran berhasil ditambahkan dan bisa tampil dihalaman utama dan halaman penyaluran.
10	Admin	Pengujian input Program.	Menambahkan Program berupa (zakat, infaq dan beasiswa).	Program berhasil ditambahkan dan bisa tampil dihalaman utama dan halaman program.
11	Admin	Pengujian konfirmasi pembayaran	Mendapatkan informasi transaksi yang dikirimkan user.	Infomasi berhasil didapatkan dengan detail-detail transaksi.
12	Admin	Pengujian konfirmasi pembayaran	Mendapat informasi bukti pembayaran dari user.	Bukti pembayaran berhasil dilihat dari hasil kiriman user.
13	Admin	Pengujian konfirmasi pembayaran	Mendapat detail pembayaran dari user yang menunggu verifikasi.	Admin dapat berhasil mengkonfirmasi ataupun membatalkan transaksi dari user.
14	Admin	Pengujian Report pembayaran user.	Hasil laporan transaksi user.	Admin bisa melihat detail transaksi yang dilakukan user dan berhasil mendapatkan total dari semua transaksi user.

III.5 Tempat dan Waktu Penelitian

A. Tempat Penelitian

Kebutuhan Penelitian ini dilakukan di *Islamic Trust Fund* yang berada di Universitas Islam Negeri Ar-Raniry Banda Aceh, Jl. Syeikh Abdul Rauf Kopelma Darussalam, Banda Aceh.

B. Waktu Penelitian

Waktu penelitian pada penelitian ini dimulai pada 22 Februari 2024 dan masih berlanjut hingga sampai dengan hari ini.



BAB IV

HASIL DAN PEMBAHASAN

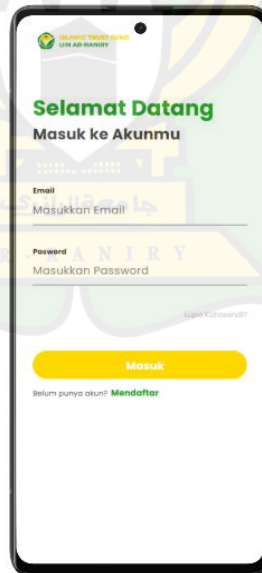
IV.1 Implementasi

Implementasi merupakan tahap kritis dalam proses pengembangan sistem yang bertujuan untuk menerapkan dan mengoperasikan sistem yang telah dirancang. Pada penelitian ini, implementasi dilakukan untuk menerapkan aplikasi *Mobile Islamic Trust Fund (ITF)* yang telah dirancang pada bab sebelumnya.

IV.1.1 Fitur Untuk *User*

Fitur untuk *User* pada aplikasi ITF dirancang untuk memudahkan pengguna dalam mengelola akun, membayar zakat, berinfaq, mendaftar beasiswa, serta memantau riwayat transaksi dan penyaluran dana. Fitur ini mencakup kalkulator zakat, pengaturan profil, serta akses ke informasi tentang ITF.

1. *Login*



Gambar IV.1 Halaman *Login*

Pada Gambar IV.1 *Login* adalah proses autentikasi yang memungkinkan pengguna untuk mengakses akun mereka dalam aplikasi dengan memasukkan kredensial seperti email dan kata sandi. Dalam konteks aplikasi *Islamic Trust Fund* (ITF), *Login* memastikan bahwa hanya pengguna yang sah dapat mengakses fitur-fitur aplikasi, seperti pembayaran zakat dan manajemen profil, serta melindungi data pengguna dari akses yang tidak sah.

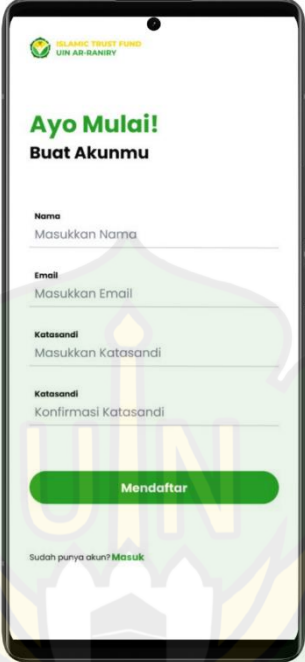
```
login() async {
  if (katasandi.text != "" && email.text != "") {
    try {
      await FirebaseAuth.instance.signInWithEmailAndPassword(
        email: email.text, password: katasandi.text);
      Navigator.pushReplacement(context, MaterialPageRoute(builder: (context) => bottomBar()));
      normalizeData();
    } on FirebaseAuthException catch (e) {
      return ScaffoldMessenger.of(context).showSnackBar(SnackBar(
        content: Text(
          '${e.message}',
          style: TextStyle(color: Colors.white),
        ), // Text
        duration: Duration(milliseconds: 500),
        backgroundColor: Colors.red,
      )); // SnackBar
    }
  } else {
    return ScaffoldMessenger.of(context).showSnackBar(SnackBar(
      content: Text(
        'Mohon isi seluruh kolom',
        style: TextStyle(color: Colors.white),
      ), // Text
      duration: Duration(milliseconds: 300),
      backgroundColor: Colors.red,
    )); // SnackBar
  }
}
```

Gambar IV.2 Program Halaman *Login*

Kode yang terdapat pada Gambar IV.2 merupakan fungsi `login()` untuk autentikasi pengguna menggunakan *Firebase Authentication*. Fungsi ini memeriksa apakah kolom input email dan kata sandi telah terisi; jika tidak, pesan peringatan ditampilkan menggunakan `SnackBar`. Ketika input valid, fungsi mencoba *Login* dengan `signInWithEmailAndPassword`. Jika berhasil, pengguna diarahkan ke halaman utama aplikasi dengan `Navigator.pushReplacement`, dan data tambahan diproses melalui `normalizeData()`. Jika terjadi kesalahan, seperti email atau kata sandi salah, `FirebaseAuthException` menangkap error dan menampilkan pesan detailnya kepada pengguna melalui `SnackBar` dengan warna merah. Fungsi ini memastikan *Login* yang aman, informatif, dan responsif.

2. Registrasi

Pada Gambar IV.3 *Registrasi* adalah proses pendaftaran akun baru di aplikasi, di mana pengguna memasukkan informasi seperti email dan kata sandi untuk mengakses fitur-fitur aplikasi secara aman.



ISLAMIC RELIEF FUND
UIN AR-RANIRY

Ayo Mulai! Buat Akunmu

Nama
Masukkan Nama

Email
Masukkan Email

Katasandi
Masukkan Katasandi

Katasandi
Konfirmasi Katasandi

Mendaftar

Sudah punya akun? [Masuk](#)

Gambar IV.3 Halaman *Register*

```

mendaftarAkun() async {
  if (nama.text != "" &&
      email.text != "" &&
      katasandi.text != "" &&
      katasandiConFirm.text != "") {
    if (katasandi.text == katasandiConFirm.text) {
      try {
        await FirebaseAuth.instance.createUserWithEmailAndPassword(
          email: email.text, password: katasandi.text);
        //upload data user
        await Databaservice(Uid: FirebaseAuth.instance.currentUser!.uid)
          .uploadDataPengguna(nama.text, email.text, "");
        await FirebaseAuth.instance.signOut();
        Navigator.pop(context);
        normalizeData();

        return ScaffoldMessenger.of(context).showSnackBar(SnackBar(
          content: Text(
            'Pendaftaran berhasil, silahkan Masukkan kembali data anda',
            style: TextStyle(color: Colors.white),
          ), // Text
          duration: Duration(seconds: 2),
          backgroundColor: const Color.fromARGB(255, 44, 144, 47),
        )); // SnackBar
      } on FirebaseAuthException catch (e) {
        return ScaffoldMessenger.of(context).showSnackBar(SnackBar(
          content: Text(
            '${e.message}',
            style: TextStyle(color: Colors.white),
          ), // Text
          duration: Duration(milliseconds: 500),
          backgroundColor: Colors.red,
        )); // SnackBar
      }
    } else {
      return ScaffoldMessenger.of(context).showSnackBar(SnackBar(
        content: Text(
          'Kata sandi tidak sama',
          style: TextStyle(color: Colors.white),
        ), // Text
        duration: Duration(milliseconds: 300),
        backgroundColor: Colors.red,
      )); // SnackBar
    }
  } else {
    return ScaffoldMessenger.of(context).showSnackBar(SnackBar(
      content: Text(
        'Mohon isi seluruh kolom',
        style: TextStyle(color: Colors.white),
      ), // Text
      duration: Duration(milliseconds: 300),
      backgroundColor: Colors.red,
    )); // SnackBar
  }
}

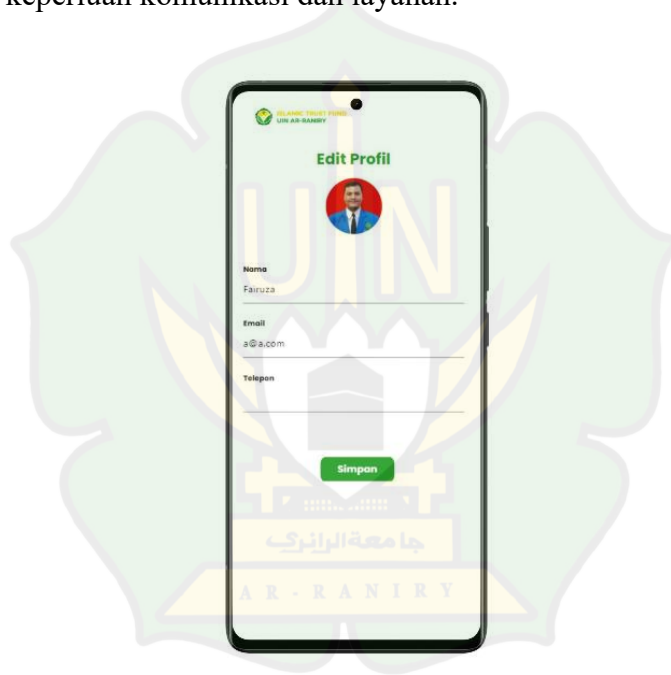
```

Gambar IV.4 Program Halaman Register

Kode yang terdapat pada Gambar IV.4 adalah fungsi `mendaftarAkun()` yang digunakan untuk registrasi pengguna baru menggunakan `Firebase Authentication`. Fungsi ini memvalidasi apakah semua kolom input (nama, email, kata sandi, dan konfirmasi kata sandi) telah diisi, serta memastikan kata sandi dan konfirmasi kata sandi sesuai. Jika valid, fungsi membuat akun baru menggunakan `createUserWithEmailAndPassword`, menyimpan data pengguna ke database dengan `Firestore.instance.currentUser!.uid`, dan menampilkan pesan sukses melalui `SnackBar`. Setelah itu, pengguna diarahkan kembali ke halaman sebelumnya. Jika terjadi error seperti dari `Firestore` atau kata sandi tidak cocok, pesan error yang sesuai ditampilkan menggunakan `SnackBar`. Fungsi ini memastikan proses registrasi berjalan dengan aman, terorganisir, dan memberikan umpan balik yang jelas kepada pengguna.

3. Edit Profil

Fitur Edit Profil, seperti yang ditampilkan pada Gambar IV.5, memungkinkan pengguna untuk memperbarui informasi pribadi mereka secara mudah dan cepat. Melalui fitur ini, pengguna dapat mengedit nama, email, dan nomor telepon sesuai kebutuhan. Semua perubahan yang dilakukan akan langsung disimpan di sistem, memastikan data pengguna tetap akurat dan terkini. Fitur ini dirancang dengan antarmuka yang sederhana dan intuitif, sehingga pengguna dapat memperbarui informasi mereka tanpa kesulitan. Dengan adanya fitur ini, aplikasi ITF mendukung personalisasi dan fleksibilitas, serta memastikan data pengguna dikelola secara efisien untuk keperluan komunikasi dan layanan.



Gambar IV.5 Halaman Edit Profil

```
Future editDataPengguna(String username, String email, String telepon) async {  
  await FirebaseFirestore.instance.collection("user").doc(Uid).update({  
    "username": username,  
    "email": email,  
    "telepon": telepon,  
  });  
}
```

Gambar IV.6 Program Halaman Edit Profil

Pada Gambar IV.6, kode tersebut adalah fungsi *editDataPengguna()* yang digunakan untuk memperbarui data pengguna di database *Firebase Firestore*. Fungsi ini menerima tiga parameter, yaitu *`username`*, *`email`*, dan *`telepon`*, lalu menggunakan *`update`* untuk memperbarui dokumen pada koleksi *`user`* berdasarkan *`Uid`* (ID pengguna unik). Data yang diperbarui adalah nama pengguna, email, dan nomor telepon. Fungsi ini berjalan secara asinkron untuk memastikan proses pembaruan selesai sebelum melanjutkan operasi lain. Dengan fungsi ini, data pengguna dapat diperbarui secara dinamis di *Firestore*.

4. Beasiswa

Fitur Menu Program untuk Daftar Beasiswa, seperti yang ditampilkan pada Gambar IV.7, dirancang untuk memberikan akses mudah bagi pengguna yang ingin mendaftar ke program beasiswa yang disediakan oleh ITF. Dalam menu ini, pengguna dapat melihat informasi lengkap tentang program beasiswa, seperti persyaratan pendaftaran, dokumen yang dibutuhkan, dan jadwal seleksi. Pengguna dapat langsung mengisi formulir pendaftaran melalui aplikasi dengan mengunggah dokumen yang diperlukan dan melengkapi data diri. Selain itu, fitur ini juga memungkinkan pengguna untuk memantau status pendaftaran mereka secara *real-time*. Dengan antarmuka yang intuitif, fitur ini membantu meningkatkan partisipasi pengguna dalam program beasiswa, sekaligus mendukung transparansi dan efisiensi proses seleksi yang dilakukan oleh ITF.



Gambar IV.7 Halaman Beasiswa

```
// URL yang akan dibuka
final String url;

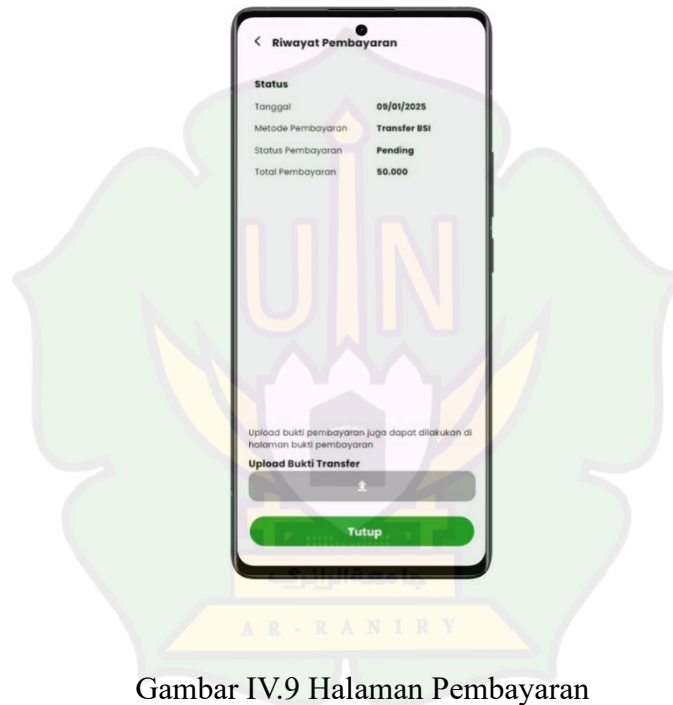
// Fungsi untuk membuka URL
Future jalankan() async {
  if (await canLaunch(url)) {
    await launch(url);
  } else {
    throw 'Tidak dapat membuka URL: $url';
  }
}
```

Gambar IV.8 Program Halaman Beasiswa

Pada Gambar IV.8 kode ini adalah fungsi `jalankan()` yang digunakan untuk membuka sebuah *URL* menggunakan *library* seperti `URL_launcher` di *Flutter*. Fungsi ini memeriksa apakah *URL* dapat diluncurkan dengan `canLaunch(URL)`, dan jika valid, *URL* akan dibuka menggunakan `launch(URL)`. Jika *URL* tidak valid atau tidak dapat dibuka, fungsi akan melempar exception dengan pesan "Tidak dapat membuka *URL*: [*URL*]". Fungsi ini memastikan *URL* yang diberikan dapat dibuka sebelum mencoba meluncurkannya, sehingga mencegah error saat runtime.

5. Pembayaran

Fitur Pembayaran pada gambar IV.9 merupakan sebuah layanan dalam aplikasi yang memungkinkan pengguna untuk melakukan pembayaran zakat atau memberikan infaq secara mudah dan cepat. Fitur ini biasanya terhubung dengan sistem pembayaran digital yang aman, seperti transfer bank atau dompet digital, untuk mendukung transaksi. Pengguna dapat memilih jenis zakat atau infaq yang akan dibayarkan, mengisi nominal sesuai kebutuhan, dan menyelesaikan pembayaran secara langsung melalui platform.



Gambar IV.9 Halaman Pembayaran

```

Future pembayaranPegguna(
    int bayaran, String tipe, String status, String tanggal, String jenis) async {
    int x = 0;
    int y = 0;
    String namaPegguna = '';
    await FirebaseFirestore.instance
        .collection("user")
        .doc(Uid)
        .collection("riwayat_pembayaran")
        .get()
        .then((value) {
            value.docs.forEach((action) {
                if (int.parse(action.id) >= x) {
                    x = int.parse(action.id) + 1;
                }
            });
        });
    await FirebaseFirestore.instance
        .collection("user")
        .doc(Uid)
        .get()
        .then((value) {
            namaPegguna = value['username'];
        });
    await FirebaseFirestore.instance
        .collection("user")
        .doc("rNgzPv2qWgNA7BDpOPSeIqhAW6P2")
        .collection("proses_pembayaran")
        .set()
        .then((value) {
            value.docs.forEach((action) {
                if (int.parse(action.id) >= y) {
                    y = int.parse(action.id) + 1;
                }
            });
        });
    await FirebaseFirestore.instance
        .collection("user")
        .doc(Uid)
        .collection("riwayat_pembayaran")
        .doc(x.toString())
        .set({
            "tanggal": tanggal,
            "bayaran": bayaran,
            "tipe": tipe,
            "status": status,
            "id_pengguna": Uid,
            "id_pembayaran": x.toString(),
            "urlBuktiBayar": "",
            "jenisPembayaran": jenis,
        });
}

```

```

await FirebaseFirestore.instance
    .collection("user")
    .doc("rNgzPv2qWgNA7BDpOPSeIqhAW6P2")
    .collection("proses_pembayaran")
    .doc(y.toString())
    .set({
        "tanggal": tanggal,
        "bayaran": bayaran,
        "tipe": tipe,
        "status": status,
        "nama_pengguna": namaPegguna,
        "id_pengguna": Uid,
        "id_pembayaran": x.toString(),
        "urlBuktiBayar": "",
        "jenisPembayaran": jenis,
    });

```

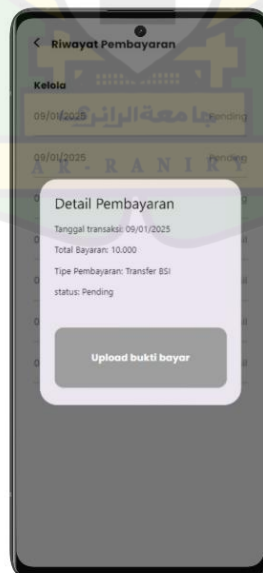
Gambar IV.10 Program Halaman Pembayaran

Pada Gambar IV.10, kode ini adalah fungsi `pembayaranPegguna()` yang bertujuan untuk mencatat transaksi pembayaran (seperti zakat atau infaq) ke dalam database *Firestore*. Fungsi ini menerima beberapa parameter, termasuk nominal pembayaran, tipe transaksi, status, tanggal, dan jenis pembayaran. Pertama, fungsi menghitung ID transaksi terbaru dari riwayat pembayaran yang ada untuk memastikan ID unik. Selanjutnya, fungsi mengambil nama pengguna berdasarkan ID pengguna (*Uid*) yang sedang *Login*. Kemudian, data pembayaran disimpan ke dua koleksi, yaitu *riwayat_pembayaran* untuk mencatat riwayat

pengguna dan `proses_pembayaran` untuk mencatat transaksi yang sedang diproses. Data yang disimpan mencakup detail transaksi, seperti tanggal, jumlah pembayaran, tipe, status, ID pengguna, dan jenis pembayaran. Fungsi ini memastikan pencatatan transaksi berjalan terstruktur dan dapat diakses oleh pengguna serta *Admin*.

6. *Upload* Bukti Pembayaran

Fitur *Upload* Pembayaran zakat/Infaq pada gambar IV.11 dirancang untuk memudahkan pengguna dalam mengunggah bukti pembayaran setelah melakukan transaksi zakat atau infaq. Fitur ini biasanya memungkinkan pengguna untuk mengisi informasi pembayaran, seperti jenis pembayaran, jumlah yang dibayarkan, dan detail lainnya, disertai dengan opsi untuk mengunggah bukti pembayaran berupa gambar. Setelah bukti pembayaran diunggah, data akan disimpan ke dalam sistem untuk diverifikasi oleh *Admin* atau pihak yang bertanggung jawab. Fitur ini memberikan transparansi dalam transaksi zakat dan infaq, memastikan pembayaran tercatat dengan baik, dan memberikan pengalaman yang *user-friendly* dalam menunaikan kewajiban berzakat atau bersedekah.



Gambar IV.11 Halaman *Upload* Bukti Pembayaran

```

UploadGambar() async {
  //cek data lagak that
  try {
    //upload ke firebase storage
    ref = FirebaseStorage.instance
      .ref()
      .child('berita')
      .child('/' + filePilihan!.files.first.name);

    final metadata = SettableMetadata(contentType: 'image/jpeg');

    uploadTask = ref!.putData(filePilihan!.files.first.bytes!, metadata);

    await uploadTask!.whenComplete(() => null);
  } on FirebaseException catch (e) {
    print(e);
  }
}

```

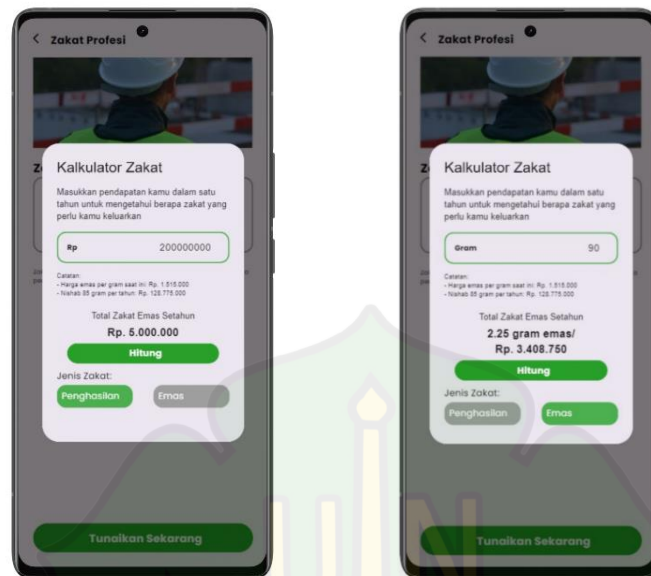
Gambar IV.12 Program Halaman *Upload* Bukti Pembayaran

Pada Gambar IV.12, kode ini adalah fungsi `UploadGambar()` yang digunakan untuk mengunggah *file* gambar ke *Firestore* Storage. Fungsi ini membuat referensi ke lokasi penyimpanan di *Firestore* Storage dengan menentukan folder dan nama *file* berdasarkan *file* yang dipilih pengguna. Metadata *file*, seperti jenis konten (`image/jpeg`), ditentukan menggunakan `SettableMetadata`. *File* yang dipilih kemudian diunggah ke *Firestore* Storage menggunakan `putData`, yang mengambil *byte* data dari *file* tersebut bersama metadatanya. Proses pengunggahan ditunggu hingga selesai dengan `whenComplete`, dan jika terjadi kesalahan selama proses, error ditangkap dengan `on FirebaseException catch` dan dicetak ke konsol. Fungsi ini memungkinkan pengguna mengunggah gambar secara mudah dan terorganisir ke *Firestore*.

7. Kalkulator Zakat Penghasilan dan Emas

Fitur Kalkulator Zakat Penghasilan dan Emas, seperti yang ditampilkan pada Gambar IV.13, dirancang untuk membantu pengguna menghitung kewajiban zakat mereka dengan mudah dan akurat. Pengguna cukup memasukkan data seperti jumlah penghasilan atau nilai emas yang dimiliki, dan kalkulator ini secara otomatis menghitung besaran zakat yang harus dikeluarkan sesuai dengan ketentuan syariah. Fitur ini juga mencantumkan nishab sebagai acuan perhitungan zakat, sehingga memastikan perhitungan dilakukan dengan tepat. Dengan antarmuka yang intuitif,

fitur ini mempermudah pengguna dalam memahami kewajiban zakat mereka dan langsung menyalurkannya melalui aplikasi, menjadikan proses pengelolaan zakat lebih praktis dan efisien.



Gambar IV.13 Halaman Kalkulator Zakat Penghasilan & Emas

```

ontap: () {
  pendapatanPengguna =
    int.parse(
      pendapatanPengguna_text
        .text);
  if (isPenghasilan) {
    if (pendapatanPengguna <
      nishab) {
      totalZakat =
        'Tidak Wajib Zakat';
    } else {
      totalZakat =
        'Rp. ${NumberFormat('#,###', 'id_ID').format(pendapatanPengguna * 2.5 / 100)}';
    }
    setState(
      () {},
    );
  } else {
    if (pendapatanPengguna <
      nishabEmas) {
      totalZakat =
        'Tidak Wajib Zakat';
    } else {
      totalZakat =
        '${pendapatanPengguna * 2.5 / 100} gram emas\nRp. ${NumberFormat('#,###', 'id_ID').format((pendapatanPengguna * 2.5 / 100) * 1515000)}';
    }
    setState(
      () {},
    );
  }
}

```

Gambar IV.14 Program Halaman Kalkulator Zakat Penghasilan & Emas

Pada Gambar IV.14, kode ini adalah logika perhitungan zakat berdasarkan pendapatan pengguna, yang dipicu ketika sebuah tombol ditekan (*onTap*). Pertama, pendapatan pengguna diambil dari input teks dan dikonversi menjadi angka. Kemudian, kode memeriksa jenis zakat (misalnya zakat penghasilan atau zakat emas) menggunakan variabel *isPenghasilan*. Jika zakat penghasilan, pendapatan dibandingkan dengan *nishab* untuk menentukan apakah pengguna wajib membayar zakat. Jika wajib, zakat dihitung sebesar 2.5% dari pendapatan dan diformat menggunakan *NumberFormat* untuk tampilan yang rapi. Jika zakat emas, pendapatan dibandingkan dengan *nishabEmas*, dan jika wajib, jumlah zakat dihitung dalam bentuk gram emas dan dikonversi ke rupiah dengan nilai emas saat ini. Hasil perhitungan ditampilkan dalam *totalZakat*, dan state aplikasi diperbarui dengan *setState* untuk merefleksikan perubahan. Fungsi ini mempermudah pengguna dalam menghitung kewajiban zakat mereka secara otomatis.

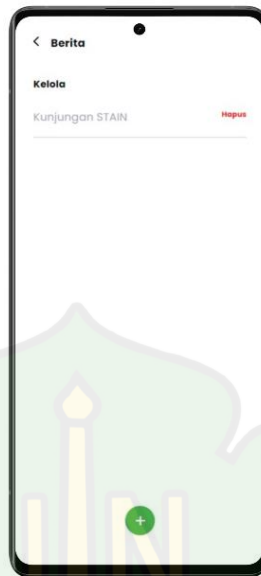
IV.1.2 Fitur Untuk Admin

Fitur untuk *Admin* pada aplikasi ITF dirancang untuk mendukung pengelolaan layanan ZISWAF secara efisien dan terorganisir. *Admin* dapat mengelola data pengguna, memverifikasi pembayaran zakat, infaq, dan donasi, serta memantau dan melaporkan penyaluran dana. Selain itu, *Admin* memiliki akses untuk mengelola program-program yang tersedia, seperti zakat, beasiswa, atau sedekah, termasuk memperbarui informasi dan kategori program. Dengan fitur ini, *Admin* dapat memastikan bahwa semua aktivitas operasional berjalan transparan, sesuai dengan prinsip syariah, dan memberikan dampak maksimal bagi penerima manfaat.

A. Tambah Berita

Fitur Tambah Berita pada mode *Admin*, seperti yang ditampilkan pada Gambar IV.15, memungkinkan *Admin* untuk menambahkan informasi terbaru ke dalam aplikasi ITF. Melalui fitur ini, *Admin* dapat membuat berita dengan mengisi judul, isi berita, dan mengunggah gambar pendukung. Fitur ini dirancang untuk mempermudah *Admin* dalam menyampaikan informasi penting, seperti laporan penyaluran dana, pengumuman program baru, atau perkembangan kegiatan ITF, kepada pengguna. Dengan antarmuka yang sederhana dan fungsional, fitur ini

memastikan bahwa berita yang ditambahkan oleh *Admin* dapat langsung ditampilkan kepada pengguna, sehingga menjaga transparansi dan keterlibatan pengguna dengan aktivitas ITF.



Gambar IV.15 Halaman Tambah Berita

```
Future tambahBerita(String judul, String deskripsi, String urlImage) async {
  int x = 0;
  await FirebaseFirestore.instance.collection("berita").get().then((value) {
    value.docs.forEach((action) {
      if (int.parse(action.id) >= x) {
        x = int.parse(action.id) + 1;
      }
    });
  });

  await FirebaseFirestore.instance
    .collection("berita")
    .doc(0.toString())
    .set({
      "judul": judul,
      "urlGambar": urlImage,
      "deskripsi": deskripsi,
    });
}
```

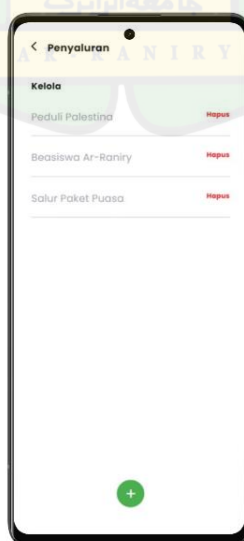
Gambar IV.16 Program Halaman Tambah Berita

Pada Gambar IV.16, kode ini adalah fungsi `tambahBerita()` yang digunakan untuk menambahkan berita baru ke *Firebase Firestore*. Fungsi

menerima tiga parameter: `'judul'`, `'deskripsi'`, dan `'URLImage'` (*URL gambar*). Pertama, fungsi mengambil data koleksi `'berita'` untuk mencari ID berita terakhir, lalu menentukan ID baru dengan menambah 1 dari ID tertinggi yang ada. Setelah itu, berita baru disimpan ke koleksi `'berita'` dengan dokumen yang berisi judul, *URL gambar*, dan deskripsi yang diberikan. Fungsi ini memastikan setiap berita memiliki ID unik dan memungkinkan penambahan berita secara terstruktur ke dalam database.

B. Tambah Penyaluran

Fitur Tambah Penyaluran pada mode *Admin*, seperti yang ditampilkan pada Gambar IV.17, memungkinkan *Admin* untuk menambahkan informasi detail terkait penyaluran zakat, infaq, atau donasi lainnya ke dalam aplikasi ITF. Melalui fitur ini, *Admin* dapat memasukkan data seperti nama program, jumlah dana yang disalurkan, penerima manfaat, dalam deskripsi penyaluran. Fitur ini dirancang untuk memastikan transparansi dalam pengelolaan dana dengan memberikan pengguna akses langsung ke informasi penyaluran terbaru. Dengan antarmuka yang *user-friendly*, fitur ini membantu *Admin* dalam mengelola dan memperbarui laporan penyaluran secara efisien, sehingga meningkatkan kepercayaan pengguna terhadap pengelolaan dana yang dilakukan oleh ITF.



Gambar IV.17 Halaman Tambah Penyaluran

```

Future tambahPenyaluran(
    String judul, String deskripsi, String urlImage) async {
    int x = 0;
    await FirebaseFirestore.instance
        .collection("penyaluran")
        .get()
        .then((value) {
            value.docs.forEach((action) {
                if (int.parse(action.id) >= x) {
                    x = int.parse(action.id) + 1;
                }
            });
        });
    await FirebaseFirestore.instance
        .collection("penyaluran")
        .doc(x.toString())
        .set({
            "judul": judul,
            "urlGambar": urlImage,
            "deskripsi": deskripsi,
        });
}

```

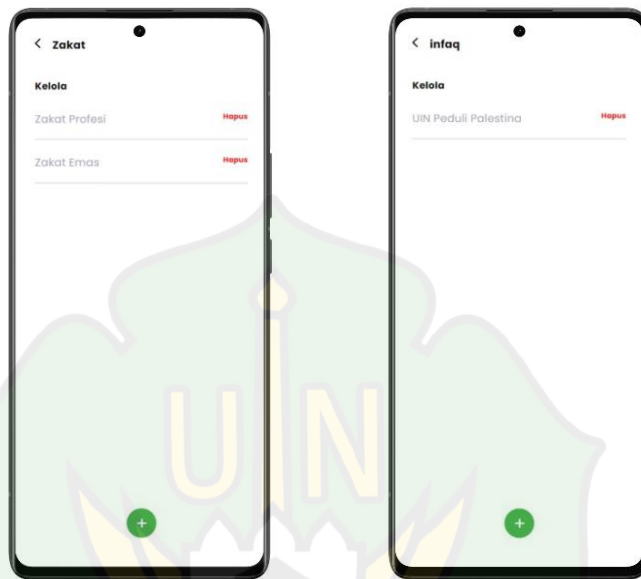
Gambar IV.18 Program Halaman Tambah Penyaluran

Pada Gambar IV.18, kode ini adalah fungsi `tambahPenyaluran()` yang digunakan untuk menambahkan data penyaluran baru ke *Firestore*. Fungsi ini menerima tiga parameter: `judul`, `deskripsi`, dan `URLImage` (*URL* gambar). Pertama, fungsi mengambil data dari koleksi `penyaluran` untuk mencari ID penyaluran terakhir, lalu menentukan ID baru dengan menambah 1 dari ID tertinggi yang ada. Setelah ID baru ditentukan, data penyaluran baru, yang mencakup judul, deskripsi, dan *URL* gambar, disimpan ke dalam koleksi `penyaluran` di *Firestore* dengan dokumen yang memiliki ID unik. Fungsi ini memastikan penambahan data penyaluran berjalan secara terstruktur dan dapat dikelola dengan baik.

C. Tambah Zakat dan Infaq

Fitur Tambah Program Zakat dan Infaq pada mode *Admin*, seperti yang ditampilkan pada Gambar IV.19, memungkinkan *Admin* untuk menambahkan program baru ke dalam aplikasi ITF. Fitur ini dirancang untuk memudahkan *Admin* dalam mengelola program-program zakat dan infaq dengan menambahkan informasi seperti nama program, deskripsi, kategori, dan target penerima manfaat.

Admin juga dapat memperbarui daftar program sesuai dengan kebutuhan atau kampanye yang sedang berjalan. Dengan antarmuka yang intuitif, fitur ini memastikan bahwa program-program baru dapat diakses pengguna secara langsung, sehingga mendorong partisipasi lebih besar dan mendukung pengelolaan ZISWAF yang lebih terstruktur dan efektif.



Gambar IV.19 Halaman Tambah Zakat dan Infaq

```
Future tambahZakat(String judul, String deskripsi, String urlImage,
String title_desc) async {
  int x = 0;
  await FirebaseFirestore.instance.collection("program").get().then((value)
  value.docs.forEach((action) {
    if (int.parse(action.id) >= x) {
      x = int.parse(action.id) + 1;
    }
  });
  await FirebaseFirestore.instance
    .collection("program")
    .doc(x.toString())
    .set({
      "judul": judul,
      "urlGambar": urlImage,
      "deskripsi": deskripsi,
      "judulDeskripsi": title_desc,
      "tipe": "zakat"
    });
}
```

Gambar IV.20 Program Halaman Tambah Zakat

Pada Gambar IV.20, kode ini adalah fungsi `tambahZakat()` yang digunakan untuk menambahkan data program zakat baru ke *Firestore*. Fungsi ini menerima parameter: `judul`, `deskripsi`, `URLImage`, dan `title_desc` (judul tambahan untuk deskripsi). Pertama, fungsi mengambil data dari koleksi `program` untuk mencari ID program terakhir, lalu menentukan ID baru dengan menambah 1 dari ID tertinggi yang ada. Setelah ID baru ditentukan, data program zakat, termasuk judul, deskripsi, *URL* gambar, judul deskripsi, dan tipe ("zakat"), disimpan ke koleksi `program` dalam dokumen dengan ID unik. Fungsi ini mempermudah penambahan program zakat secara terorganisir dalam database *Firestore*.

```
Future tambahInfaq(String judul, String deskripsi, String urlImage,
String title_desc) async {
  int x = 0;
  await FirebaseFirestore.instance.collection("program").get().then((value) {
    value.docs.forEach((action) {
      if (int.parse(action.id) >= x) {
        x = int.parse(action.id) + 1;
      }
    });
  });
  await FirebaseFirestore.instance
    .collection("program")
    .doc(x.toString())
    .set({
      "judul": judul,
      "urlGambar": urlImage,
      "deskripsi": deskripsi,
      "judulDeskripsi": title_desc,
      "tipe": "infaq"
    });
}
```

Gambar IV.21 Program Halaman Tambah Infaq

Pada Gambar IV.21, kode ini adalah fungsi `tambahInfaq()` yang digunakan untuk menambahkan data program infaq baru ke *Firestore*. Fungsi ini menerima parameter: `judul`, `deskripsi`, `URLImage`, dan `title_desc` (judul tambahan untuk deskripsi). Pertama, fungsi mengambil data dari koleksi `program` untuk mencari ID program terakhir, lalu menentukan ID baru dengan menambah 1 dari ID tertinggi yang ada. Setelah ID baru ditentukan, data program infaq, termasuk judul, deskripsi, *URL* gambar, judul deskripsi, dan tipe ("infaq"), disimpan ke koleksi `program` dalam dokumen dengan ID unik. Fungsi ini

memastikan program infaq dapat ditambahkan ke dalam database *Firestore* secara rapi dan terstruktur.

D. Beasiswa

Fitur Tambah Beasiswa pada mode *Admin*, seperti yang ditampilkan pada Gambar IV.22, dirancang untuk mempermudah *Admin* dalam menambahkan program beasiswa baru ke dalam aplikasi ITF. Melalui fitur ini, *Admin* dapat memasukkan informasi penting seperti nama program beasiswa, deskripsi, persyaratan, dan periode pendaftaran. Dengan antarmuka yang sederhana, fitur ini membantu *Admin* mengatur dan memperbarui daftar beasiswa secara efisien.



Gambar IV.22 Halaman Tambah Beasiswa

```

Future tambahbeasiswa(String judul, String deskripsi, String urlImage,
String title_desc, String url) async {
  int x = 0;
  await FirebaseFirestore.instance.collection("program").get().then((value) {
    value.docs.forEach((action) {
      if (int.parse(action.id) >= x) {
        x = int.parse(action.id) + 1;
      }
    });
  });
  await FirebaseFirestore.instance
    .collection("program")
    .doc(x.toString())
    .set({
      "judul": judul,
      "urlGambar": urlImage,
      "deskripsi": deskripsi,
      "judulDeskripsi": title_desc,
      "url": url,
      "tipe": "beasiswa"
    });
}

```

Gambar IV.23 Program Halaman Tambah Beasiswa

Pada Gambar IV.23, kode ini adalah fungsi `tambahBeasiswa()` yang digunakan untuk menambahkan data program beasiswa baru ke *Firestore*. Fungsi ini menerima parameter: `judul`, `deskripsi`, `URLImage`, `title_desc`, dan `URL`. Pertama, fungsi mengambil data dari koleksi `program` untuk mencari ID program terakhir, lalu menentukan ID baru dengan menambah 1 dari ID tertinggi yang ada. Setelah ID baru ditentukan, data program beasiswa, termasuk judul, deskripsi, URL gambar, judul deskripsi, URL terkait, dan tipe ("beasiswa"), disimpan ke koleksi `program` dalam dokumen dengan ID unik. Fungsi ini memastikan program beasiswa ditambahkan secara terorganisir ke dalam database *Firestore*, sehingga memudahkan pengelolaan data.

E. Konfirmasi Pembayaran

Fitur Menu Konfirmasi pada mode *Admin*, seperti yang ditampilkan pada Gambar IV.24, dirancang untuk mempermudah *Admin* dalam memverifikasi pembayaran zakat dan infaq yang dilakukan oleh pengguna. Melalui fitur ini, *Admin* dapat melihat daftar transaksi yang memerlukan konfirmasi, termasuk detail seperti nama pengguna, jumlah pembayaran, metode pembayaran, dan waktu transaksi. *Admin* dapat menyetujui atau menolak transaksi berdasarkan validasi yang dilakukan. Fitur ini memastikan bahwa setiap pembayaran yang masuk tercatat dengan akurat dan transparan, serta memungkinkan dana yang telah terkonfirmasi

segera dikelola atau disalurkan sesuai dengan program yang dituju. Dengan antarmuka yang intuitif, fitur ini mendukung efisiensi pengelolaan dana ZISWAF oleh *Admin*.



Gambar IV.24 Halaman Konfirmasi Pembayaran

```
Future konfirmasiPembayaran(String id_pengguna, String id_pembayaranPengguna,
    String id_pembayaran) async {
    await FirebaseFirestore.instance
        .collection("user")
        .doc(Uid)
        .collection("proses_pembayaran")
        .doc(id_pembayaran)
        .delete();
    await FirebaseFirestore.instance
        .collection("user")
        .doc(id_pengguna)
        .collection("riwayat_pembayaran")
        .doc(id_pembayaranPengguna)
        .update({
            "status": "Berhasil",
        });
}
```

Gambar IV.25 Program Halaman Konfirmasi Pembayaran

Pada Gambar IV.25, kode ini adalah fungsi `konfirmasiPembayaran()` yang digunakan untuk memproses dan mengonfirmasi pembayaran pengguna di *Firestore*. Fungsi ini menerima tiga parameter: `id_pengguna`, `id_pembayaran`,

dan `id_pembayaranPegguna`. Pertama, fungsi menghapus data pembayaran dari koleksi `proses_pembayaran`, yang digunakan untuk menyimpan pembayaran yang sedang menunggu konfirmasi. Selanjutnya, fungsi memperbarui status pembayaran di koleksi `riwayat_pembayaran` pengguna yang relevan menjadi "Berhasil". Dengan ini, pembayaran yang telah dikonfirmasi akan dipindahkan dari daftar proses ke riwayat, memberikan status yang jelas kepada pengguna dan *Admin* mengenai transaksi tersebut.

F. *Report Pembayaran User*

Fitur Menu *Report Pembayaran User* pada mode *Admin*, seperti yang ditampilkan pada Gambar IV.26, dirancang untuk memberikan *Admin* akses ke laporan lengkap semua transaksi yang dilakukan oleh pengguna. Melalui fitur ini, *Admin* dapat melihat riwayat transaksi pengguna secara detail, termasuk nama pengguna, jenis pembayaran (zakat atau infaq), jumlah transaksi, tanggal, dan status pembayaran. Selain itu, fitur ini juga menyajikan total akumulasi semua transaksi yang telah dilakukan oleh pengguna dalam periode tertentu. Dengan antarmuka yang terstruktur, fitur ini memudahkan *Admin* dalam memantau aktivitas keuangan aplikasi, membuat laporan, dan memastikan transparansi dalam pengelolaan dana. Fitur ini menjadi alat penting untuk evaluasi dan pelaporan operasional ITF.



Gambar IV.26 Halaman *Report Pembayaran User*


```

getdata() async {
  seluruhPembayaranZakat.clear();
  seluruhPembayaranInfaq.clear();
  totalTransaksi = 0;
  await FirebaseFirestore.instance
    .collection("user")
    .doc(widget.id)
    .collection("riwayat_pembayaran")
    .get()
    .then((value) {
      value.docs.forEach((action) {
        if (action['jenisPembayaran'] == 'zakat' && action['status'] == 'Berhasil') {
          seluruhPembayaranZakat.add(dataPembayaranZakat());
          seluruhPembayaranZakat[seluruhPembayaranZakat.length - 1].id =
            action.id;
          seluruhPembayaranZakat[seluruhPembayaranZakat.length - 1].status =
            action['status'];
          seluruhPembayaranZakat[seluruhPembayaranZakat.length - 1].tanggal =
            action['tanggal'];
          seluruhPembayaranZakat[seluruhPembayaranZakat.length - 1].tipe =
            action['tipe'];
          seluruhPembayaranZakat[seluruhPembayaranZakat.length - 1]
            .id_pengguna = action['id_pengguna'];
          seluruhPembayaranZakat[seluruhPembayaranZakat.length - 1].bayaran =
            action['bayaran'];
          seluruhPembayaranZakat[seluruhPembayaranZakat.length - 1]
            .id_pembayaranPengguna = action['id_pembayaran'];
          seluruhPembayaranZakat[seluruhPembayaranZakat.length - 1]
            .urlBuktiBayar = action['urlBuktiBayar'];
        }
      });
    });
}
});

```

```

await FirebaseFirestore.instance
  .collection("user")
  .doc(widget.id)
  .collection("riwayat_pembayaran")
  .get()
  .then((value) {
    value.docs.forEach((action) {
      if (action['jenisPembayaran'] == 'infaq' && action['status'] == 'Berhasil') {
        seluruhPembayaranInfaq.add(dataPembayaranInfaq());
        seluruhPembayaranInfaq[seluruhPembayaranInfaq.length - 1].id =
          action.id;
        seluruhPembayaranInfaq[seluruhPembayaranInfaq.length - 1].status =
          action['status'];
        seluruhPembayaranInfaq[seluruhPembayaranInfaq.length - 1].tanggal =
          action['tanggal'];
        seluruhPembayaranInfaq[seluruhPembayaranInfaq.length - 1].tipe =
          action['tipe'];
        seluruhPembayaranInfaq[seluruhPembayaranInfaq.length - 1]
          .id_pengguna = action['id_pengguna'];
        seluruhPembayaranInfaq[seluruhPembayaranInfaq.length - 1].bayaran =
          action['bayaran'];
        seluruhPembayaranInfaq[seluruhPembayaranInfaq.length - 1]
          .id_pembayaranPengguna = action['id_pembayaran'];
        seluruhPembayaranInfaq[seluruhPembayaranInfaq.length - 1]
          .urlBuktiBayar = action['urlBuktiBayar'];
      }
    });
  });
  for (int i = 0; i < seluruhPembayaranZakat.length; i++) {
    totalTransaksi += seluruhPembayaranZakat[i].bayaran;
  }
  for (int i = 0; i < seluruhPembayaranInfaq.length; i++) {
    totalTransaksi += seluruhPembayaranInfaq[i].bayaran;
  }
}

```

Gambar IV.27 Program Halaman *Report Pembayaran User*

Pada Gambar IV.27, kode ini adalah fungsi `getdata()` yang bertugas mengambil data riwayat pembayaran zakat dan infaq dari *Firestore*, memisahkan keduanya, dan menghitung total transaksi. Pertama, fungsi membersihkan list `seluruhPembayaranZakat` dan `seluruhPembayaranInfaq`, lalu mengatur `totalTransaksi` menjadi nol. Data diambil dari koleksi `riwayat_pembayaran` pengguna berdasarkan `widget.id`. Data pembayaran yang berhasil (`status` = "Berhasil") disortir berdasarkan jenisnya, yaitu zakat atau infaq. Setiap data ditambahkan ke list yang sesuai dengan rincian seperti ID, status, tanggal, tipe, ID pengguna, jumlah pembayaran, dan *URL* bukti pembayaran. Setelah itu, total transaksi dihitung dengan menjumlahkan pembayaran zakat dan infaq secara terpisah. Fungsi ini memastikan data pembayaran pengguna ditampilkan terorganisir dan terhitung dengan benar.

IV.2 Pengujian Sistem

Pengujian sistem merupakan proses untuk menguji aplikasi yang telah dikembangkan guna memastikan bahwa seluruh fitur yang dirancang berfungsi dengan baik sesuai dengan harapan. Pengujian ini bertujuan untuk mengidentifikasi kekurangan pada aplikasi dengan mengevaluasi setiap fitur, sehingga dapat ditemukan bug yang menghambat aplikasi bekerja secara optimal atau bahkan

menyebabkan aplikasi tidak dapat digunakan oleh pengguna. Dalam pengujian ini, metode yang digunakan adalah *black box testing*.

Black box testing bertujuan untuk menguji seluruh fitur yang tersedia tanpa memperhatikan desain maupun kode program dari aplikasi. Hal ini dilakukan untuk memastikan bahwa aplikasi dapat memenuhi kebutuhan pengguna sebagaimana yang diharapkan.

Tabel IV.1 Hasil Pengujian *Black Box*

No	Aktor	Skenario Pengujian	Uji Kasus	Hasil Yang Diharapkan	Kesimpulan
1	<i>User</i>	Pengujian fungsi <i>Registrer</i> .	Mendaftarkan nama, email, katasandi.	Akun berhasil terbuat.	Berhasil
2	<i>User</i>	Pengujian fungsi <i>Login</i> .	Mengisi email dan katasandi yang benar.	Sistem melakukan <i>Login</i> , <i>user</i> masuk ke halaman utama.	Berhasil
3	<i>User</i>	Pengujian fungsi <i>Login</i> .	Mengisi email dan katasandi yang salah.	Sistem akan mengirimkan pesan kesalahan.	Berhasil
4	<i>User</i>	Pengujian pembayaran zakat & infaq.	Melakukan transaksi pembayaran dan meng <i>Upload</i> bukti pembayaran.	Pembayaran bisa selesai terlaksana dan bisa meng <i>Upload</i> bukti pembayaran sampai terverifikasi dari <i>Admin</i> .	Berhasil
5	<i>User</i>	Pengujian riwayat pembayaran	Melihat status dari transaksi yang dilakukan <i>user</i> .	Informasi transaksi bisa terlihat, <i>user</i> dapat melihat status pembayaran dan bukti transfer yang pernah di <i>Upload</i> dan juga bisa meng <i>Upload</i> bukti pembayaran jika sebelumnya belum di <i>Upload</i> .	Berhasil

No	Aktor	Skenario Pengujian	Uji Kasus	Hasil Yang Diharapkan	Kesimpulan
6	<i>User</i>	Pengujian kalkulator zakat.	Mengisi sejumlah bilangan pada kalkulator untuk menghitung jumlah zakat yang dikeluarkan.	Kalkulator berhasil menghitung jumlah zakat yang perlu dikeluarkan oleh <i>user</i> .	Berhasil
7	<i>User</i>	Pengujian Beasiswa	Memilih beasiswa yang diinginkan.	Beasiswa berhasil dipilih dan <i>user</i> diarahkan ke google form untuk mengisi formulir beasiswa yang dipilih.	Berhasil
8	<i>Admin</i>	Pengujian input Berita.	Menambahkan Berita.	Berita berhasil ditambahkan dan bisa tampil dihalaman utama.	Berhasil
9	<i>Admin</i>	Pengujian input Penyaluran.	Menambahkan Penyaluran.	Penyaluran berhasil ditambahkan dan bisa tampil dihalaman utama dan halaman penyaluran.	Berhasil
10	<i>Admin</i>	Pengujian input Program.	Menambahkan Program berupa (zakat, infaq dan beasiswa).	Program berhasil ditambahkan dan bisa tampil dihalaman utama dan halaman program.	Berhasil
11	<i>Admin</i>	Pengujian konfirmasi pembayaran	Mendapatkan informasi transaksi yang dikirimkan <i>user</i> .	Infomasi berhasil didapatkan dengan detail-detail transaksi.	Berhasil
12	<i>Admin</i>	Pengujian konfirmasi pembayaran	Mendapat informasi bukti pembayaran dari <i>user</i> .	Bukti pembayaran berhasil dilihat dari hasil kiriman <i>user</i> .	Berhasil
13	<i>Admin</i>	Pengujian konfirmasi pembayaran	Mendapat detail pembayaran dari <i>user</i> yang	<i>Admin</i> dapat berhasil mengkonfirmasi	Berhasil

			menunggu verifikasi.	ataupun membatalkan transaksi dari <i>user</i> .	
No	Aktor	Skenario Pengujian	Uji Kasus	Hasil Yang Diharapkan	Kesimpulan
14	<i>Admin</i>	Pengujian <i>Report</i> pembayaran <i>user</i> .	Hasil laporan transaksi <i>user</i> .	<i>Admin</i> bisa melihat detail transaksi yang dilakukan <i>user</i> dan berhasil mendapatkan total dari semua transaksi <i>user</i> .	Berhasil



BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Penelitian ini berhasil membangun sebuah aplikasi *mobile* untuk *Islamic Trust Fund* (ITF) UIN Ar-Raniry Banda Aceh yang dirancang guna meningkatkan efisiensi, transparansi, dan akuntabilitas pengelolaan dana Zakat, Infaq, Shadaqah, Wakaf (ZISWAF). Dengan fitur-fitur utama seperti pembayaran zakat, pembayaran infaq, beasiswa, kalkulator zakat, riwayat transaksi, laporan penyaluran, aplikasi ini memberikan kemudahan kepada pengguna dalam menunaikan kewajiban zakat. Selain itu, sistem pengelolaan berbasis digital yang diterapkan memungkinkan *Admin* untuk mengelola program dengan lebih terorganisir. Pengujian menggunakan metode *black box* menunjukkan bahwa aplikasi telah memenuhi kebutuhan fungsional dengan hasil yang sesuai harapan. Implementasi aplikasi ini memberikan solusi modern terhadap pengumpulan dana ZISWAF.

5.2 Saran

Disarankan agar aplikasi ITF ini dikembangkan lebih lanjut dengan menambahkan fitur interaktif, seperti notifikasi pengingat jadwal pembayaran zakat dan konsultasi zakat dengan ahli, untuk meningkatkan keterlibatan pengguna. Selain itu, integrasi dengan metode pembayaran digital lainnya, seperti *e-wallet* atau *QRIS*, dapat memperluas fleksibilitas pembayaran. Pengujian performa aplikasi juga perlu diperluas dengan berbagai skenario pengguna untuk memastikan stabilitas sistem, terutama pada kondisi beban tinggi. Dari sisi keamanan, implementasi enkripsi end-to-end dan autentikasi dua faktor sangat dianjurkan untuk melindungi data pengguna dengan lebih baik. Agar pemanfaatan aplikasi ini optimal, pihak ITF juga disarankan melakukan sosialisasi dan pelatihan kepada pengguna dan *Admin* terkait penggunaan fitur-fitur yang tersedia. Dengan pengembangan dan langkah-langkah ini, aplikasi diharapkan dapat memberikan dampak lebih besar dalam mendukung pengelolaan ZISWAF secara modern dan efisien.

DAFTAR PUSTAKA

- Zidan, M., Nur'aini, S., Wibowo, N. C. H., & Ulinuha, M. A. (2022). *Black box Testing pada Aplikasi Single Sign On (SSO) di Diskominfo Standi Menggunakan Teknik Equivalence Partitions*. *Walisongo Journal of Information Technology*, 4(2), 127–137. <https://doi.org/10.21580/wjit.2022.4.2.12135>
- Aceng Abdul Wahid. (2020). *Analisis Metode Waterfall Untuk Pengembangan Sistem Informasi*. <https://www.researchgate.net/publication/346397070>
- Agus Muhyidin, M., Sulhan, M. A., & Sevtiana, A. (2020). *Perancangan Ui/Ux Aplikasi My Cic Layanan Informasi Akademik Mahasiswa Menggunakan Aplikasi Figma* (Vol. 10, Issue 2). <https://my.cic.ac.id/>.
- Alamsyah, M., & Firman Arif, M. (2019). PENDATAAN PENERIMA ZAKAT KANTOR BAZNAS KABUPATEN PROBOLINGGO BERBASIS ANDROID GIS. In *JOUTICA* (Vol. 4, Issue 2). <http://pusat.baznas.go.id/profil/>
- Alfarisi, S. (2023). *Perancangan Aplikasi Pembayaran Zakat Pada Yayasan Baituzzakah Berbasis Android*.
- Enggar Krisnada, F., & Tanone, R. (2020). Aplikasi Penjualan Tiket Kelas Pelatihan Berbasis *Mobile* menggunakan *Flutter*. *Jurnal Teknik Informatika Dan Sistem Informasi*, 5(3). <https://doi.org/10.28932/jutisi.v5i3.1865>
- Firmansyah, Y., & Febryansyah, R. (n.d.). PENERAPAN METODE *WATERFALL* UNTUK APLIKASI *MOBILE* PADA ARUS KAS MASJID DI LINGKUNGAN KOTA METRO. In *Teknologipintar.org* (Vol. 1, Issue 1).
- Hanif, I. F., & Sinambela, G. M. (2020). *PEMBUATAN APLIKASI E-TATIB BERBASIS ANDROID MENGGUNAKAN BAHASA PEMROGRAMAN DART MAKING AN ANDROID-BASED E-TATIB APPLICATION USING THE DART PROGRAMMING LANGUAGE*. 3(2), 23–29.
- Imron, I., Azizah, N., Nurhayati, M. S., & Wijonarko, B. (2021). Perancangan Aplikasi *Mobile* Zakat dan Infaq Berbasis Android Pada Baznas Kabupaten Tangerang. *Jurnal Ilmiah Universitas Batanghari Jambi*, 21(1), 197. <https://doi.org/10.33087/jiubj.v21i1.1234>
- Panjaitan, J., & Pakpahan, A. F. (2021). Perancangan Sistem *E-Reporting* Menggunakan ReactJS dan *Firebase*. *Jurnal Teknik Informatika Dan Sistem Informasi*, 7(1). <https://doi.org/10.28932/jutisi.v7i1.3098>

- Romzi, M., & Kurniawan, B. (2020). Implementasi Pemrograman Python Menggunakan Visual Studio Code. In *JIK: Vol. XI* (Issue 2). www.python.org
- Sahibul Nuzul Firdaus. (2023). *PENGEMBANGAN APLIKASI KAMUS BAHASA ACEH BERBASIS MOBILE*. UIN Ar-Raniry Banda Aceh.
- Sugita, A., Rohmat Hidayat, A., Hardiyanto, F., & Wulandari, S. I. (2020). Analisis Peranan Pengelolaan Dana Ziswaf Dalam Pemberdayaan Ekonomi Umat Pada Lazisnu Kabupaten Cirebon. *Jurnal Indonesia Sosial Sains*, 1(1), 9–18. <https://doi.org/10.36418/jiss.v1i1.6>

