

**SISTEM INFORMASI PENDAFTARAN SANTRI BARU
BERBASIS MOBILE DENGAN PUSH NOTIFICATION
MENGUNAKAN FIREBASE CLOUD MESSAGING (FCM)
DI PONDOK PESANTREN AL-FALAH MOGA**

SKRIPSI

Diajukan untuk Memenuhi Sebagian Syarat Guna
Memperoleh Gelar Sarjana Strata 1 (S1)
dalam Ilmu Teknologi Informasi



Diajukan oleh :

MUHAMMAD DHIYAUL ALLAM

NIM : 1908096047

**PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI WALISONGO SEMARANG**

2024

PERNYATAAN KEASLIAN

Yang bertanda tangan di bawah ini:

Nama : Muhammad Dhiyaul Allam

NIM : 1908096047

Jurusan : Teknologi Informasi

Menyatakan bahwa skripsi yang berjudul :

**SISTEM INFORMASI PENDAFTARAN SANTRI BARU
BERBASIS MOBILE DENGAN PUSH NOTIFICATION
MENGUNAKAN FIREBASE CLOUD MESSAGING (FCM) DI
PONDOK PESANTREN AL-FALAH MOGA**

Secara keseluruhan adalah hasil penelitian/karya saya sendiri,
kecuali bagian tertentu yang dirujuk sumbernya.

Semarang, 20 Juni 2024

Pembuat Pernyataan,



Muhammad Dhiyaul Allam

NIM. 1908096047



KEMENTERIAN AGAMA REPUBLIK INDONESIA
UNIVERSITAS ISLAM NEGERI WALISONGO SEMARANG
FAKULTAS SAINS DAN TEKNOLOGI

Jl. Prof. Dr. Hamka Km. 1 Semarang Telp. 024 76433366 Semarang 50185
E-mail: fst@walisongo.ac.id Web : <http://fst.walisongo.ac.id>

PENGESAHAN

Naskah skripsi berikut ini:

Judul : Sistem Informasi Pendaftaran Santri Baru Berbasis
Mobile dengan Push Notification Menggunakan
Firebase Cloud Messaging (FCM) di Pondok
Pesantren Al-Falah Moga
Penulis : Muhammad Dhiyaul Allam
NIM : 1908096047
Jurusan : Teknologi Informasi

Telah diujikan dalam sidang tugas akhir oleh Dewan Penguji
Fakultas Sains dan Teknologi UIN Walisongo dan dapat
diterima sebagai salah satu syarat memperoleh gelar sarjana
dalam Teknologi Informasi.

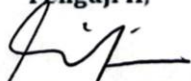
Semarang, Juni 2024

DEWAN PENGUJI

Penguji I,


Nur Cahyo Hendro, S.T., M.Kom.
NIP. 197312222006041001

Penguji II,


Dr. Khotibul Umam, S.T., M.Kom.
NIP. 197908272011011007

Penguji III,


Hery Mustofa, M.Kom.
NIP. 198703172019031001

Penguji IV,


Mokhammad Ikhsan Mustofa, M.Kom.
NIP. 198808072019031010

Pembimbing I,


Dr. Khotibul Umam, S.T., M.Kom.
NIP. 197908272011011007

Pembimbing II,


Siti Nur'aini, M.Kom.
NIP. 199107032019031006

NOTA PEMBIMBING

Yth. Ketua Program Studi Teknologi Informasi
Fakultas Sains dan Teknologi
UIN Walisongo Semarang

Assalamu'alaikum wr.wb.

Dengan ini diberitahukan bahwa saya telah melakukan bimbingan, arahan dan koreksi naskah skripsi dengan :

Judul : Sistem Informasi Pendaftaran Santri Baru
Berbasis Mobile dengan Push Notification
Menggunakan Firebase Cloud Messaging (FCM)
di Pondok Pesantren Al-Falah Moga

Penulis : Muhammad Dhiyaul Allam

NIM : 1908096047

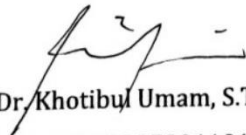
Jurusan : Teknologi Informasi

Saya memandang bahwa naskah skripsi tersebut sudah dapat diajukan kepada Fakultas Sains dan Teknologi UIN Walisongo untuk diujikan dalam Sidang Munaqosyah.

Wassalamu'alaikum wr.wb.

Semarang, 20 Juni 2024

Pembimbing I,


Dr. Khotibul Umam, S.T, M.Kom
NIP. 197908272011011007

NOTA PEMBIMBING

Yth. Ketua Program Studi Teknologi Informasi
Fakultas Sains dan Teknologi
UIN Walisongo Semarang

Assalamu'alaikum wr.wb.

Dengan ini diberitahukan bahwa saya telah melakukan bimbingan, arahan dan koreksi naskah skripsi dengan :

Judul : Sistem Informasi Pendaftaran Santri Baru
Berbasis Mobile dengan Push Notification
Menggunakan Firebase Cloud Messaging
(FCM) di Pondok Pesantren Al-Falah Moga

Penulis : Muhammad Dhiyaul Allam

NIM : 1908096047

Jurusan : Teknologi Informasi

Saya memandang bahwa naskah skripsi tersebut sudah dapat diajukan kepada Fakultas Sains dan Teknologi UIN Walisongo untuk diujikan dalam Sidang Munaqosyah.

Wassalamu'alaikum wr.wb.

Semarang, 20 Juni 2024

Pembimbing II,



Siti Nur'aini, M.Kom

NIP. 199107032019031006

ABSTRAK

Pondok Pesantren Al-Falah Moga merupakan salah satu pondok pesantren yang terletak di Kabupaten Pemalang. Proses pendaftaran santri baru di Pondok Pesantren Al-Falah Moga masih dilakukan secara manual dan belum maksimal. Untuk meningkatkan efisiensi dan transparansi proses pendaftaran santri baru, diperlukan integrasi teknologi informasi berupa sistem informasi pendaftaran santri baru berbasis *mobile* dengan *push notification* menggunakan Firebase Cloud Messaging (FCM), sehingga dapat memberikan kemudahan akses dan penyampaian informasi secara cepat dan tepat. Sistem informasi ini dirancang menggunakan metode pengembangan sistem Software Development Lifecycle (SDLC) model *waterfall* berupa analisis kebutuhan, perancangan sistem, implementasi, pengujian sistem, serta pengoperasian dan perbaikan. Hasil pengujian sistem menggunakan metode *black box testing* menunjukkan bahwa aplikasi dapat menjalankan fungsi pendaftaran serta *push notification* yang memberikan informasi secara akurat dan tepat sasaran.

Kata kunci: Sistem Informasi Pendaftaran, Aplikasi *Mobile*, Firebase Cloud Messaging

KATA PENGANTAR

Dengan memanjatkan puji dan syukur kehadirat Allah SWT yang telah melimpahkan segala rahmat hidayah serta inayah-Nya sehingga dapat terselesaikan laporan tugas akhir skripsi dengan judul “Sistem Informasi Pendaftaran Santri Baru Berbasis *Mobile* dengan *Push Notification* Menggunakan Firebase Cloud Messaging (FCM) di Pondok Pesantren Al-Falah Moga”.

Adapun penyusunan skripsi ini tidak selalu berjalan dengan mulus dan lancar sehingga banyak pihak yang terlibat dalam penyelesaiannya. Oleh karena itu, penulis hendak menyampaikan rasa terima kasih yang sebesar-besarnya kepada:

1. Bapak Prof. Dr. Nizar, M.Ag., Selaku Rektor UIN Walisongo Semarang.
2. Bapak Dr. Khotibul Umam, S.T., M.Kom., Selaku Ketua Program Studi Teknologi Informasi UIN Walisongo Semarang dan sebagai Dosen Pembimbing I.
3. Ibu Siti Nur’aini, M.Kom., Selaku Dosen Wali dan Dosen Pembimbing II.
4. Keluarga terutama kedua orang tua penulis yang telah memberikan bantuan, dukungan serta do’a kepada penulis.

5. Segenap Dosen UIN Walisongo yang telah memberikan ilmu dan pengalaman kepada penulis.
6. Zidan, Mailan, dan Ganang yang senantiasa menemani penulis selama perkuliahan.
7. Luqman selaku teman yang telah menemani penyusunan skripsi.
8. Teman-teman mahasiswa Teknologi Informasi 2019 dan semua pihak yang telah terlibat dalam penyusunan skripsi ini sehingga dapat selesai dengan baik.

Dalam penyusunan skripsi ini, penulis menyadari bahwa penelitian ini masih terdapat berbagai kekurangan. Oleh karena itu, penulis mengharapkan adanya kritik dan saran yang bersifat membangun agar dapat menjadi bahan rujukan penulis untuk penelitian selanjutnya. Penulis berharap agar penelitian ini dapat bermanfaat dan memberikan dampak positif bagi semua pihak.

Semarang, 20 Juni 2024
Penulis,



Muhammad Dhiyaul Allam
NIM. 1908096047

DAFTAR ISI

HALAMAN JUDUL.....	i
PERNYATAAN KEASLIAN.....	ii
PENGESAHAN.....	iii
NOTA PEMBIMBING.....	iv
NOTA PEMBIMBING.....	v
ABSTRAK.....	vi
KATA PENGANTAR	vii
DAFTAR ISI	ix
DAFTAR TABEL.....	xi
DAFTAR GAMBAR.....	xii
DAFTAR LAMPIRAN.....	xiv
BAB I.....	1
PENDAHULUAN.....	1
A. Latar Belakang.....	1
B. Identifikasi Masalah.....	1
C. Rumusan Masalah.....	1
D. Batasan Masalah.....	1
E. Tujuan Penelitian	1
F. Manfaat Penelitian	1
BAB II	9
LANDASAN TEORI.....	9
A. Kajian Pustaka	9
1. Sistem Informasi	9
2. Aplikasi <i>Mobile</i>	10
3. Pondok Pesantren.....	11
4. Konsep <i>Push Notification</i>	12
5. Firebase Cloud Messaging (FCM)	16

B. Kajian Penelitian yang Relevan	19
BAB III	24
METODE PENELITIAN	24
A. Metode Pengumpulan Data	24
1. Observasi	24
2. Wawancara.....	24
3. Studi Pustaka.....	24
B. Metode Pengembangan Sistem	26
1. Analisis Kebutuhan	27
2. Perancangan Sistem	29
3. Implementasi	38
4. Pengujian Sistem.....	39
5. Pengoperasian dan Perbaikan	40
BAB IV	41
IMPLEMENTASI DAN PEMBAHASAN	41
A. Pengumpulan Data	41
1. Observasi	41
2. Wawancara.....	41
B. Implementasi Sistem	45
C. Pengujian Sistem.....	77
BAB V.....	82
KESIMPULAN DAN SARAN.....	82
DAFTAR PUSTAKA.....	84
LAMPIRAN.....	89

DAFTAR TABEL

Tabel 2.1 Kajian Penelitian Terdahulu	19
Tabel 3.1 Berkas Pendaftaran Santri Baru	28
Tabel 3.2 Pengujian <i>Blackbox Testing</i>	40
Tabel 4.1 Wawancara Penelitian	41
Tabel 4.2 Perangkat Lunak yang dipakai	46
Tabel 4.3 Perangkat Keras yang dipakai	46
Tabel 4.4 Pengujian <i>Blackbox</i> Login.....	77
Tabel 4.5 Pengujian <i>Blackbox</i> Daftar.....	78
Tabel 4.6 Pengujian <i>Blackbox</i> Form Daftar	79
Tabel 4.7 Pengujian <i>Blackbox</i> Validasi Pendaftaran.....	79
Tabel 4.8 Pengujian <i>Blackbox</i> Push Notifikasi	80

DAFTAR GAMBAR

Gambar 2.1 Alur Push Notification.....	14
Gambar 2.2 Alur Kerja FCM	17
Gambar 3.1 Alur Pendaftaran yang Sudah Berjalan	25
Gambar 3.2 Alur Pendaftaran yang diusulkan	26
Gambar 3.3 Tahapan Waterfall	27
Gambar 3.4 Use Case Diagram.....	30
Gambar 3.5 Activity Diagram.....	31
Gambar 3.6 Sequence Diagram Login	32
Gambar 3.7 Sequence Diagram Pendaftaran	33
Gambar 3.8 Sequence Diagram Pengumuman	33
Gambar 3.9 Sequence Diagram Pengiriman Notifikasi User.....	34
Gambar 3.10 Sequence Diagram Pengiriman Notifikasi Admin.....	34
Gambar 3.11 Halaman Login.....	36
Gambar 3.12 Halaman Utama User	37
Gambar 3.13 Halaman Utama Admin	38
Gambar 4.1 Database Cloud Firestore.....	47
Gambar 4.2 Halaman Daftar Akun	49
Gambar 4.3 Halaman Login Akun.....	50
Gambar 4.4 Halaman Beranda User	51
Gambar 4.5 Halaman Form Pendaftaran.....	52
Gambar 4.6 Halaman Brosur	53
Gambar 4.7 Halaman Pengumuman.....	54
Gambar 4.8 Halaman Media.....	55
Gambar 4.9 Halaman Profil User	56
Gambar 4.10 Halaman Info Aplikasi.....	57
Gambar 4.11 Halaman Beranda Admin	58
Gambar 4.12 Halaman Data Pendaftaran.....	59
Gambar 4.13 Halaman Lihat Data Pendaftaran.....	60

Gambar 4.14 Halaman Edit Brosur.....	61
Gambar 4.15 Halaman Verifikasi.....	62
Gambar 4.16 Halaman Profil Admin	63
Gambar 4.17 Notifikasi Pendaftaran Baru.....	64
Gambar 4.18 Notifikasi Saat Validasi Pendaftaran	64

DAFTAR LAMPIRAN

Lampiran 1 : Lembar Persetujuan Pembimbing	89
Lampiran 2 : Lembar Pengesahan Proposal	90
Lampiran 3 : Dokumentasi Pengujian <i>Blackbox Testing</i>	91
Lampiran 4 : Formulir Pernyataan Pendaftaran.....	92
Lampiran 5 : Daftar Riwayat Hidup.....	93

BAB I
PENDAHULUAN

BAB II

LANDASAN TEORI

A. Kajian Pustaka

1. Sistem Informasi

Sistem adalah suatu kesatuan atau kombinasi elemen-elemen yang saling berinteraksi dan saling mempengaruhi untuk mencapai suatu tujuan tertentu. (Yulianeu dan Sodik 2019) menyatakan bahwa suatu sistem dapat didefinisikan sebagai suatu kesatuan yang terdiri dari dua atau lebih komponen yang berinteraksi untuk mencapai suatu tujuan. Di dalam suatu sistem biasanya terdapat prosedur-prosedur yang saling berhubungan untuk menyelesaikan masalah guna mencapai tujuan.

Informasi adalah data yang diolah menjadi bentuk yang lebih berguna bagi penerimanya untuk mengambil keputusan saat ini atau saat mendatang (Asmara 2019). Informasi adalah data yang telah diolah dan diinterpretasikan menjadi bentuk yang berguna bagi penerima atau pengguna informasi. Informasi memberikan pemahaman dan wawasan yang lebih baik, membantu pengambilan keputusan, dan digunakan untuk tujuan-tujuan tertentu. Dalam konteks teknologi informasi, informasi sering

dikaitkan dengan data yang diolah oleh komputer atau sistem informasi.

Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengelolaan transaksi harian, mendukung operasi, bersifat manajerial, dan kegiatan strategi dari suatu organisasi tertentu dengan laporan-laporan yang dibutuhkan (Anggraini, Pasha, dan Setiawan 2020). Sistem informasi adalah suatu sistem yang terdiri dari komponen-komponen seperti orang, prosedur, perangkat keras, perangkat lunak, dan data. Tujuan dari sistem informasi adalah untuk mengumpulkan, memproses, menyimpan, dan menyebarkan informasi yang diperlukan dalam sebuah organisasi untuk membantu pengambilan keputusan yang tepat.

2. Aplikasi *Mobile*

Aplikasi adalah suatu perangkat lunak (*software*) yang dirancang untuk melakukan fungsi-fungsi tertentu atau tugas-tugas tertentu pada suatu perangkat komputer, perangkat *mobile*, atau sistem lainnya. *Mobile application* (aplikasi *mobile*) adalah sebuah bahasa pemrograman yang mempresentasikan apa yang seharusnya dilakukan oleh perangkat lunak atau bagaimana suatu proses perangkat lunak

seharusnya menyelesaikan tugasnya (Darnita dan Muntahanah 2021).

3. Pondok Pesantren

Menurut pendapat para ilmuwan, istilah pondok pesantren adalah merupakan dua istilah yang mengandung satu arti. Orang Jawa menyebutnya “pondok” atau “pesantren”. Sering pula menyebut sebagai pondok pesantren. Istilah pondok barangkali berasal dari pengertian asrama-asrama para santri yang disebut pondok atau tempat tinggal yang terbuat dari bambu atau barangkali berasal dari Bahasa Arab “funduq” artinya asrama besar yang disediakan untuk persinggahan (Kahfi dan Kasanova 2020). Menurut H. Abdul Aziz Dahlan, “Pondok pesantren adalah lembaga pendidikan yang memadukan aspek agama, moral, dan akhlak dalam rangka mencetak generasi muslim yang benar-benar menguasai ilmu pengetahuan dan teknologi, serta mampu mengaplikasikannya sesuai dengan tuntutan zaman”.

Pondok pesantren Al-Falah Moga adalah salah suatu Lembaga Pendidikan Islam yang terletak di dusun Kalibuntu Kecamatan Moga Kabupaten Pemalang. Pondok pesantren ini diasuh oleh seorang

kyai, yaitu KH. Ahmad Baidlowi Ilham Kholil (Sa'diyah 2019).

4. Konsep *Push Notification*

Notifikasi adalah pemberitahuan mengenai informasi atau pengumuman dari pihak tertentu kepada pihak yang dituju yang dilakukan melalui media seperti email, sms maupun aplikasi *chatting* (*line, whatsapp, BBM, wechat, dsb*) (Malik dan Zakaria 2023). Notifikasi sangat penting bagi penerima informasi, karena dengan notifikasi, informasi tersebut dapat langsung diketahui oleh penerima. Berikut ini adalah komponen notifikasi yang umumnya ada (Material, n.d. diakses 20 Oktober 2023):

a. Judul (*title*)

Judul merupakan bagian notifikasi yang berisi judul atau singkatan dari pesan notifikasi. Tujuannya adalah memberikan gambaran singkat tentang apa notifikasi tersebut.

b. Isi (*body*)

Bagian ini berisi isi utama dari pesan notifikasi dimana informasi atau pesan yang ingin disampaikan kepada pengguna ditampilkan. Isi notifikasi sendiri harus jelas dan informatif.

c. Gambar (*image*)

Beberapa notifikasi dapat mencakup gambar atau grafik yang relevan dengan pesan. Gambar ini bisa digunakan untuk memvisualisasikan informasi dengan lebih baik.

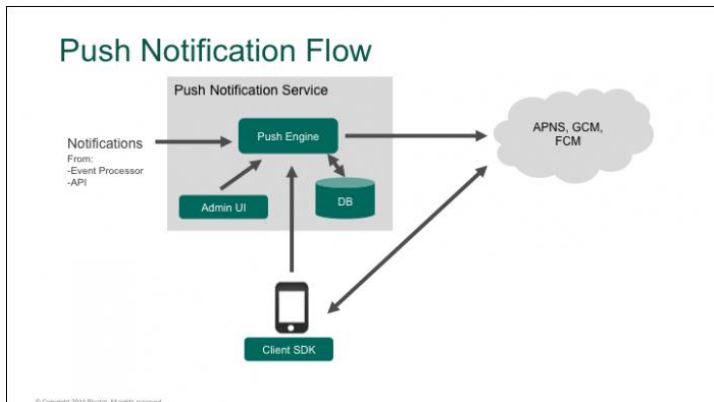
d. Tindakan (*action*)

Beberapa notifikasi dapat menyertakan tindakan yang dapat dilakukan pengguna, seperti tombol “buka” untuk membuka aplikasi terkait atau tautan ke halaman web tertentu.

Push notification adalah pesan yang “didorong” dari server *backend* atau aplikasi ke antarmuka pengguna aplikasi sasaran, namun tidak terbatas pada aplikasi seluler dan aplikasi desktop. *Push notification* dapat menargetkan perangkat tertentu ataupun sekelompok pengguna, dimana dapat bervariasi dari pesan notifikasi sederhana hingga panggilan yang tidak dapat dilihat pengguna ke *backend* aplikasi pada platform sasaran (Tommy, Wahyuningsih, dan Romadiana 2020).

Push notification termasuk ke dalam kategori dari model komunikasi internet, berdasarkan cara mempublikasikan/berlangganannya klien tidak harus menyetujui atau *merequest* dari server pusat untuk mendapatkan suatu informasi. Biasanya *push*

notification akan berbentuk pesan pendek yang muncul pada layar *smartphone* secara otomatis, walaupun pemilik *smartphone* sedang tidak membuka aplikasi tersebut (Triawan dan Siboro 2021). Bentuk dari *push notification* ini hampir mirip dengan pesan singkat atau sms, namun *push notification* biasanya hanya muncul pada *smartphone* yang telah menginstal aplikasi bersangkutan.



Gambar 2-1 Alur Push Notification (Faisol dan Rahmadianto 2018)

Ketika seorang pengguna mengunduh dan menginstal aplikasi di perangkat mereka yang mendukung *push notification*, aplikasi meminta izin dari pengguna untuk mengirim notifikasi. Setelah izin diberikan, aplikasi mengirim permintaan pendaftaran ke layanan notifikasi *push* yang sesuai, seperti Firebase Cloud Messaging untuk perangkat Android atau Apple

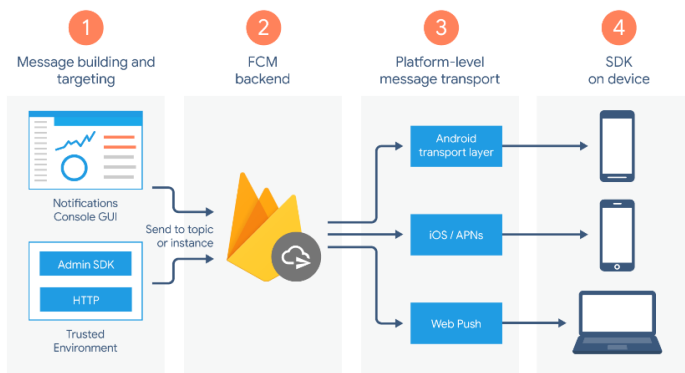
Push Notification Service untuk perangkat iOS. Pada tahap ini, setiap perangkat diberi token unik yang digunakan untuk mengidentifikasi perangkat tersebut. Layanan notifikasi *push* mendaftarkan aplikasi dan perangkat pengguna untuk menerima notifikasi. Token yang diberikan saat pendaftaran digunakan untuk menghubungkan aplikasi dengan perangkat pengguna. Ketika server aplikasi memiliki informasi atau peristiwa yang memerlukan notifikasi kepada pengguna (seperti pesan baru, pembaruan aplikasi, atau peringatan), server mengirimkan pesan notifikasi ke layanan notifikasi *push*. Pesan notifikasi berisi informasi seperti judul, isi, tindakan yang diperlukan, dan token perangkat. Kemudian layanan notifikasi *push* menggunakan token perangkat yang telah diberikan saat pendaftaran untuk mengirim notifikasi ke perangkat pengguna. Notifikasi ini mencakup teks, gambar, suara, atau elemen lain sesuai dengan format notifikasi yang telah ditentukan. Perangkat pengguna yang telah menerima notifikasi akan menampilkannya kepada pengguna. Notifikasi bisa muncul dalam bentuk pesan *pop-up*, ikon notifikasi pada layar, suara, ataupun getaran. Selain pengguna dapat melihat notifikasi yang diterima, pengguna juga dapat mengambil tindakan

yang sesuai. Mereka dapat membaca pesan, menjawab pesan, membuka aplikasi terkait dengan notifikasi, atau menekan notifikasi untuk mendapatkan informasi lebih lanjut. Interaksi pengguna dengan notifikasi tergantung pada jenis notifikasi dan tindakan yang diperlukan.

5. Firebase Cloud Messaging (FCM)

Firebase adalah suatu layanan dari Google untuk memberikan kemudahan bahkan mempermudah para *developer* aplikasi dalam mengembangkan aplikasinya. Firebase disebut juga sebagai BaaS (*backend as a service*) dan merupakan sebuah solusi yang ditawarkan oleh Google untuk mempercepat pekerjaan *developer* (Nur Anisa dan Wibowo 2022). Firebase menyediakan beragam layanan dan alat untuk membantu pengembang dalam membangun, mengelola, dan mengoptimalkan aplikasi seluler dan web mereka (Laia 2023). Ini mencakup layanan untuk autentikasi pengguna, penyimpanan data, basis data *real-time*, *hosting* web, analisis, dan tentu saja layanan *push notification* melalui Firebase Cloud Messaging (FCM). Firebase sangat populer di kalangan pengembang karena kemudahan penggunaan dan integrasinya yang kuat dengan platform Android dan iOS.

Firebase Cloud Messaging (FCM) adalah layanan *push notification* yang dikelola oleh Firebase, yang memungkinkan pengembang untuk mengirim pesan notifikasi ke perangkat Android, iOS, dan web (Somya dan Aprillia 2019). Berikut adalah penjelasan bagaimana FCM bekerja:



Gambar 2-2 Alur Kerja FCM (Firebase, n.d. diakses 20 Oktober 2023)

a. Registrasi Perangkat

Saat pengguna mengunduh dan menginstall aplikasi yang mendukung *push notification* (FCM), aplikasi tersebut meminta izin pengguna untuk menerima notifikasi. Setelah izin diberikan, aplikasi mengirimkan permintaan pendaftaran (*registration request*) ke server FCM untuk mendaftarkan perangkat pengguna.

b. Pendaftaran Aplikasi dan Perangkat

Server FCM mendaftarkan aplikasi dan perangkat pengguna dengan memberikan token unik kepada perangkat tersebut. Token inilah yang nantinya digunakan untuk mengidentifikasi perangkat secara unik dan menghubungkannya dengan aplikasi yang mengirim notifikasi.

c. Penyandian Pesan

Ketika server aplikasi memiliki informasi yang memerlukan notifikasi kepada pengguna, server tersebut menyusun pesan notifikasi dalam format yang sesuai, seperti JSON. Pesan notifikasi harus mencakup informasi seperti judul notifikasi, isi notifikasi, token perangkat yang akan menerima notifikasi, dan tindakan yang diperlukan oleh pengguna jika ada.

d. Pengiriman Pesan ke FCM

Server aplikasi mengirim pesan notifikasi ke server FCM dengan menggunakan antarmuka RESTful FCM atau SDK klien FCM yang tersedia. Pesan ini dikirim ke FCM dengan menggunakan token perangkat untuk menentukan perangkat mana yang akan menerima notifikasi.

e. Proses Pengiriman oleh FCM

Setelah menerima pesan notifikasi, server FCM mengelola pengiriman notifikasi ke perangkat yang dituju. FCM akan memeriksa token perangkat dan mencocokkannya dengan perangkat yang sesuai dalam sistemnya.

f. Penerimaan Notifikasi

Perangkat pengguna akan menerima notifikasi dari FCM dan biasanya muncul dalam bentuk pesan *pop-up*, ikon notifikasi ataupun suara tergantung pada pengaturan perangkat dan notifikasi.

g. Interaksi Pengguna

Pengguna yang mendapatkan notifikasi dapat mengambil tindakan yang sesuai, seperti membuka aplikasi terkait dengan notifikasi, membaca pesan, atau menjawab notifikasi sesuai dengan instruksi yang diberikan dalam pesan.

B. Kajian Penelitian yang Relevan

Pada tabel 2.1 di bawah ini merupakan beberapa penelitian sebelumnya yang relevan berdasarkan kesamaan topik atau pemanfaatan teknologi yang menjadi referensi dari penelitian ini, antara lain sebagai berikut:

Tabel 2.1 Kajian Penelitian Terdahulu

No	Nama Peneliti	Judul Penelitian	Hasil Penelitian
----	---------------	------------------	------------------

1	(Darmawan 2019)	Implementasi Push Notification Scheduler pada Firebase Cloud Messaging Untuk Aplikasi Pengingat Kegiatan Organisasi Menggunakan Flutter Berbasis Mobile	Hasil dari penelitian ini dapat disimpulkan bahwa aplikasi dapat menerima notifikasi dengan baik di berbagai <i>device</i> , selain itu notifikasi juga akan tetap muncul meskipun <i>device</i> dalam keadaan <i>offline</i> , dan aplikasi berhasil menerapkan <i>scheduling</i> untuk <i>push notification</i> .
2	(Meidina et al. 2022)	Penerapan Fitur Firebase Cloud Messaging pada Sistem Kontrol Pembayaran Iuran BPJS Ketenagakerjaan Berbasis <i>Mobile</i>	Penerapan fitur Firebase Cloud Messaging pada sistem dapat berjalan dengan baik, sistem berhasil mengirimkan notifikasi kepada peserta mengenai waktu jatuh tempo pembayaran iuran sehingga memudahkan peserta

			karena mendapatkan notifikasi secara <i>real-time</i> . Selain itu juga memudahkan admin dalam melakukan pendataan terhadap laporan pembayaran peserta.
3	(Nurzam, Fajri, dan Prabowo 2017)	Rancang Bangun Aplikasi Media Laporan Aspirasi dengan Firebase Cloud Messaging Berbasis Mobile	Aplikasi dapat menjalankan keseluruhan fungsi dengan baik dan dapat menuliskan laporan atau aspirasi ke pihak instansi dan <i>push notification</i> Firebase Cloud Messaging dapat dikirimkan sesuai dengan alur kerja.
4	(Cahyadi et al. 2019)	Teknologi Firebase Untuk Aplikasi Lapor Akakom	Teknologi Firebase telah berhasil diimplementasikan pada aplikasi, diantara beberapa teknologi

			yang digunakan antara lain Firebase Firestore, Firebase Cloud Messaging, Firebase Functions, Firebase Hosting dan Firebase Authentication.
5	(Faisol dan Rahmadianto 2018)	<i>Realtime Notification</i> Pada Aplikasi Berbasis Web Menggunakan Firebase Cloud Messaging (FCM)	Dengan menggunakan Firebase Cloud Messaging untuk mengirim pesan atau notifikasi secara <i>realtime</i> kepada klien, sehingga kinerja sistem dapat lebih efektif dan efisien. Selain itu, sistem juga dapat mengirim notifikasi selama <i>browser</i> terhubung dengan internet meskipun tanpa mengakses aplikasi.

Sebagai dasar perancangan, peneliti menggunakan referensi yang telah disajikan pada tabel 2.1 tersebut. Penelitian yang akan dilakukan memiliki perbedaan dengan penelitian yang telah dilakukan sebelumnya, yaitu penelitian ini akan berfokus pada sistem informasi pendaftaran santri baru dengan berbasis *mobile* yang didukung dengan penerapan *push notification* menggunakan Firebase Cloud Messaging sehingga proses pendaftaran santri baru menjadi lebih mudah dan efisien.

BAB III

METODE PENELITIAN

A. Metode Pengumpulan Data

Proses pengumpulan data yang diperlukan dalam penelitian ini menggunakan beberapa metode, antara lain:

1. Observasi

Penulis melakukan observasi atau pengamatan secara langsung di Pondok Pesantren Al-Falah Moga dengan mengamati proses pendaftaran santri baru yang berlangsung di sana.

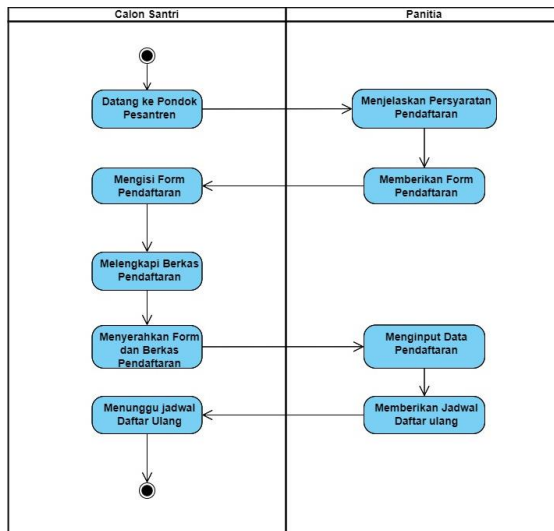
2. Wawancara

Wawancara dilakukan langsung terhadap panitia pendaftaran santri baru untuk mendapatkan data dan informasi yang akan dibutuhkan dalam pembuatan sistem informasi pendaftaran santri baru berbasis *mobile*.

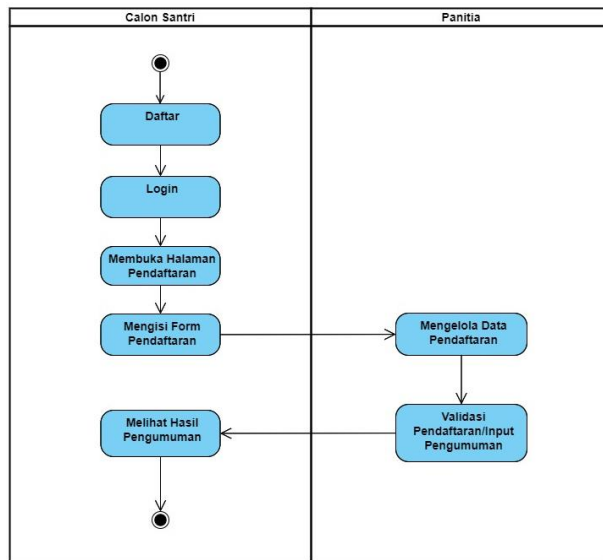
3. Studi Pustaka

Penulis melakukan studi pustaka yaitu dengan mengkaji literatur baik berupa buku, jurnal, tugas akhir ataupun sumber bacaan untuk mendapatkan referensi terkait teori-teori yang diperlukan dan relevan dalam penelitian.

Gambar berikut ini merupakan gambaran alur pendaftaran yang sudah berjalan dan gambaran alur pendaftaran yang diusulkan:



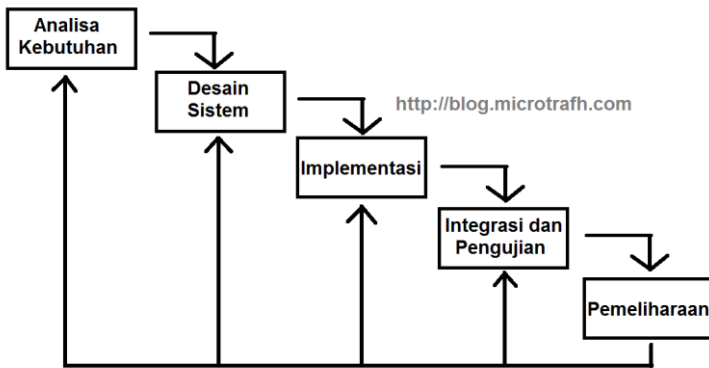
Gambar 3-1 Alur Pendaftaran yang Sudah Berjalan



Gambar 3-2 Alur Pendaftaran yang diusulkan

B. Metode Pengembangan Sistem

Metode pengembangan sistem yang digunakan dalam penelitian ini adalah Software Development Lifecycle (SDLC) model *waterfall*. SDLC merupakan sebuah proses pembuatan dan pengubahan sistem serta model dan metodologi yang digunakan untuk mengembangkan sistem-sistem tersebut (Aceng Abdul Wahid 2020). Sedangkan model *waterfall* atau biasa disebut dengan model air terjun adalah sebuah model siklus hidup klasik sistematis dalam mengembangkan sebuah *software* (perangkat lunak) (Ridwan, Fitri, dan Benrahman 2021).



Gambar 3-3 Tahapan Waterfall (Auzaiaefif19, n.d. diakses 22 Oktober 2023)

Beberapa tahapan dalam penelitian ini akan dijelaskan menggunakan model *waterfall* sebagai berikut:

1. Analisis Kebutuhan

Tahap ini digunakan untuk menganalisis kebutuhan yang diperlukan dalam pembuatan aplikasi *mobile* sistem informasi pendaftaran santri baru di Pondok Pesantren Al-Falah Moga dengan *push notification* menggunakan FCM. Beberapa kebutuhan fungsional sistem yang ada meliputi:

- a. Fitur pembagian *role* pengguna terdaftar terkait pengelolaan halaman sesuai hak aksesnya.
- b. Fitur registrasi akun bagi calon santri baru.
- c. Fitur pengisian formulir pendaftaran bagi calon santri baru.

- d. Fitur pengelolaan dan validasi data pendaftaran oleh admin.
- e. Fitur melihat hasil pengumuman pendaftaran.

Sedangkan kebutuhan non fungsional sistem dapat dilihat sebagai berikut:

- a. Pembuatan aplikasi yang mencakup pengkodean menggunakan *software* Android Studio.
- b. Bahasa pemrograman yang digunakan menggunakan bahasa Java.
- c. Sedangkan untuk *database* sistemnya, penulis menggunakan Cloud Firestore yang merupakan salah satu layanan dari Firebase.

Beberapa berkas yang diperlukan dalam formulir pendaftaran di Pondok Pesantren Al-Falah Moga dapat dilihat pada tabel 3.1 berikut ini:

Tabel 3.1 Berkas Pendaftaran Santri Baru

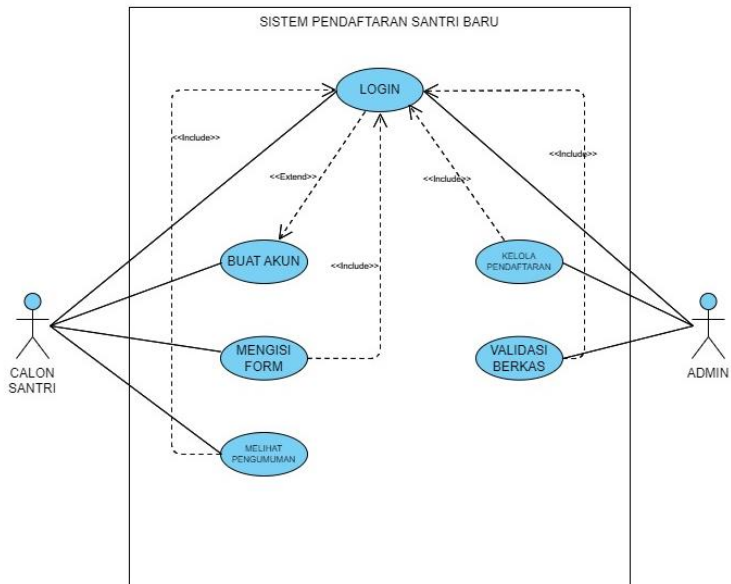
No	Berkas
1	Scan Akta Kelahiran
2	Scan Kartu Keluarga
3	Pas Foto Calon Santri
4	Scan Ijazah Madrasah

2. Perancangan Sistem

Pada tahap ini, dilakukan perancangan sistem berdasarkan analisis kebutuhan yang telah didapatkan. Tujuannya adalah agar dapat memberikan gambaran terhadap apa yang harus dikerjakan. Beberapa hal yang dihasilkan pada tahap ini antara lain sebagai berikut:

a. *Use Case Diagram*

Use case diagram merupakan langkah pertama dalam memodelkan sebuah sistem. *Use case* merupakan pemodelan untuk kebutuhan sebuah sistem fungsional, setiap *use case diagram* digambarkan sebagai kunci dari suatu skenario yang dilakukan oleh aktor dan diringkas dalam sebuah batas sistem, setiap *use case diagram* dihubungkan dengan sebuah garis notasi (Aliman 2021). Pada gambar 3.6 di bawah ini merupakan *use case diagram* kebutuhan fungsional sistem. Aktor utama dalam sistem ini adalah calon santri baru serta panitia penerimaan santri baru.

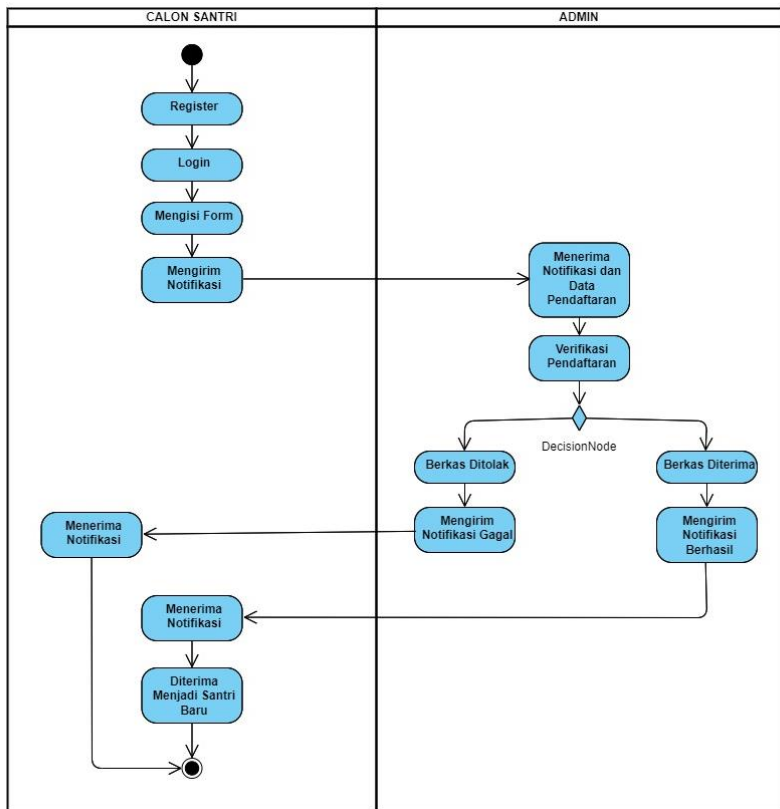


Gambar 3-4 Use Case Diagram

b. Activity Diagram

Activity diagram merupakan pemodelan yang menggambarkan sebuah sistem kerja dari sebuah objek atau sebuah sistem, sebuah *activity diagram* digambarkan dengan sebuah alur secara terstruktur proses kerja dari *use case* yang sedang diproses dari titik awal sampai titik akhir, setiap aktivitas digambarkan dengan notasi-notasi

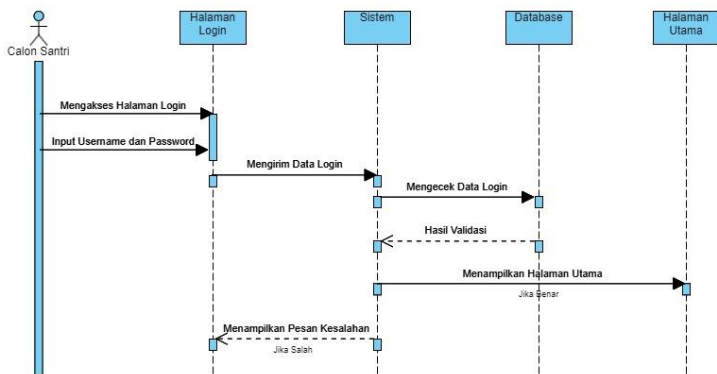
sesuai fungsinya (Aliman 2021). Berikut ini merupakan *activity diagram* yang menggambarkan alur kerja dari sistem yang akan dibangun berdasarkan *use case* yang telah dibuat.



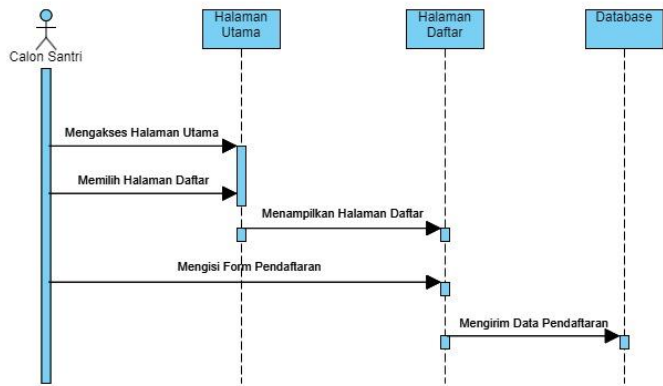
Gambar 3-5 Activity Diagram

c. Sequence Diagram

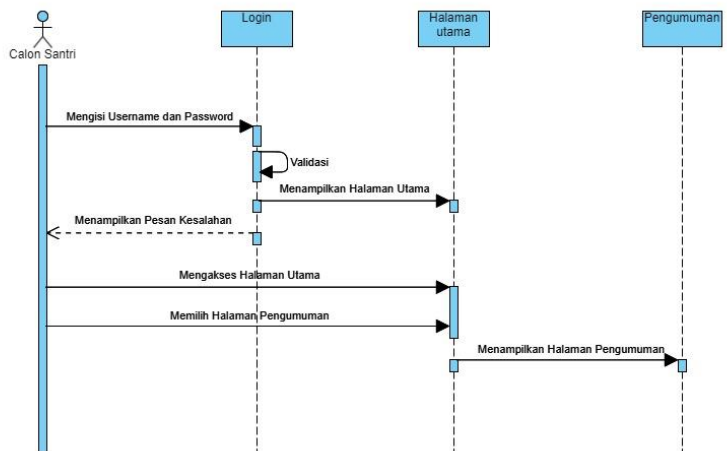
Sequence diagram merupakan UML yang menggambarkan interaksi antar objek di dalam dan di sekitar sistem, termasuk pengguna, *display*, dan sebagainya berupa *message* yang digambarkan terhadap waktu (Syarif dan Nugraha 2020).



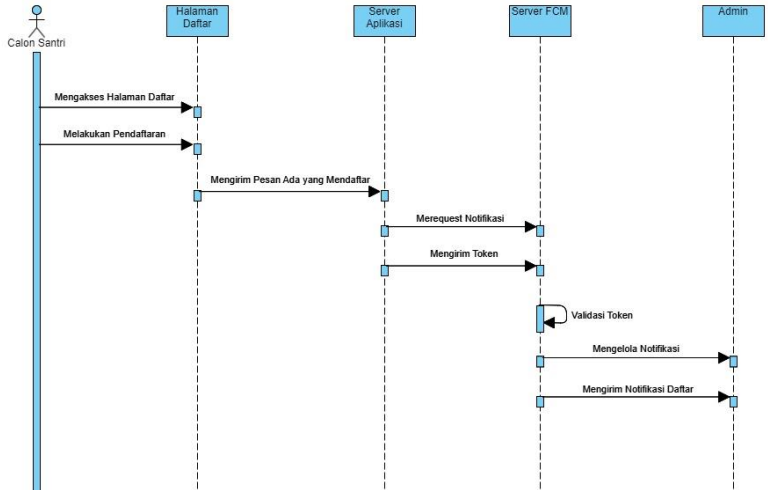
Gambar 3-6 Sequence Diagram Login



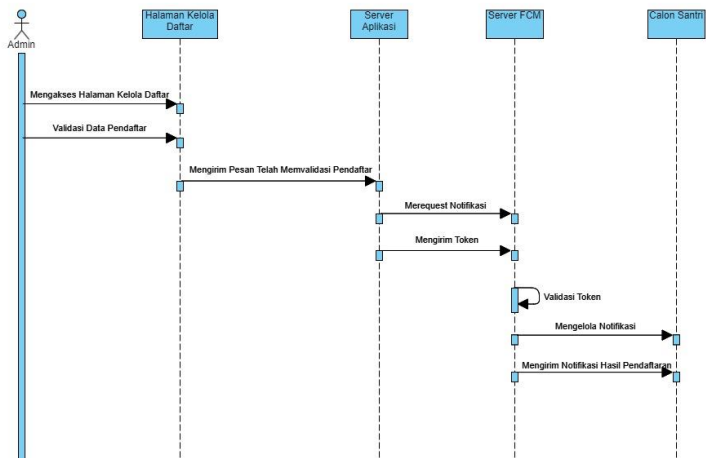
Gambar 3-7 Sequence Diagram Pendaftaran



Gambar 3-8 Sequence Diagram Pengumuman



Gambar 3-9 Sequence Diagram Pengiriman Notifikasi User



Gambar 3-10 Sequence Diagram Pengiriman Notifikasi Admin

d. *User interface*

User Interface adalah input dan output yang langsung melibatkan sistem pengguna akhir. Antarmuka pengguna dapat digunakan langsung oleh pengguna internal atau eksternal sistem. Desain dari *user interface* sendiri bisa bervariasi banyak tergantung pada faktor-faktor seperti tujuan antarmuka, karakteristik pengguna, dan karakteristik perangkat *interface* tertentu (Fernando 2020). Sebuah sistem atau aplikasi yang dapat dikatakan mudah dalam penggunaannya, ketika menggunakan desain tampilan yang baik dan juga mudah dimengerti oleh pengguna. Berikut ini adalah gambar rancangan desain *user interface* yang nantinya akan dijadikan referensi untuk perancangan aplikasi.



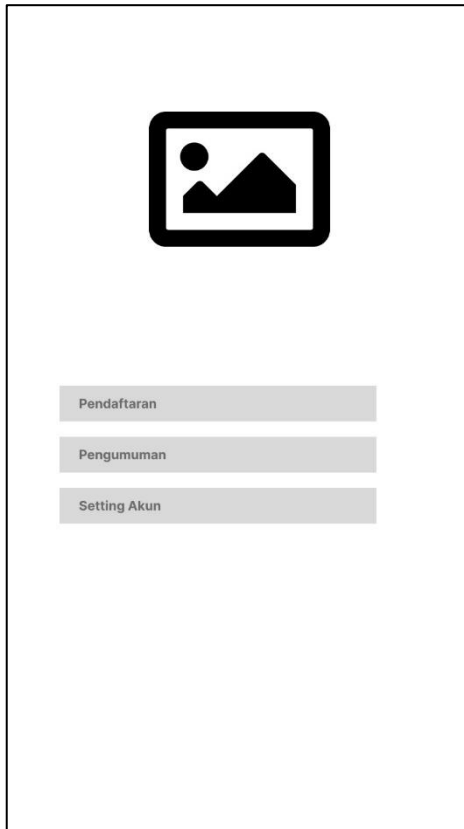
**Sistem Informasi Pendaftaran
Pondok Pesantren Al-Falah Moga**

Username

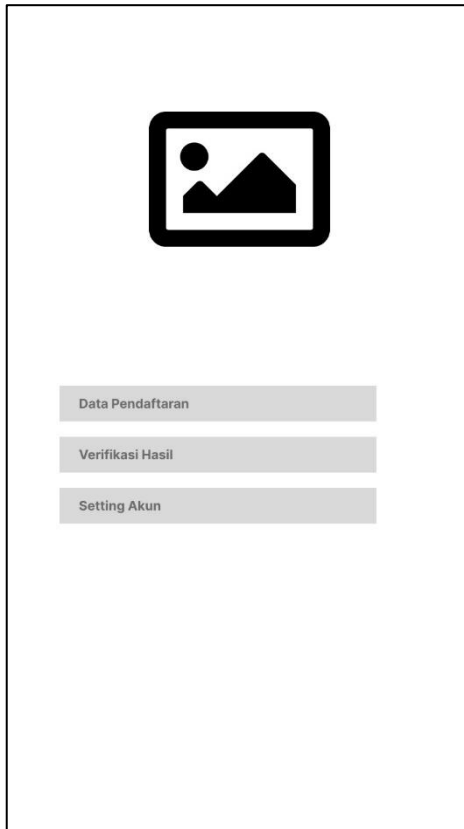
Password

Daftar **Login**

Gambar 3-11 Halaman Login



Gambar 3-12 Halaman Utama User



Gambar 3-13 Halaman Utama Admin

3. Implementasi

Pada tahap ini, desain yang telah dirancang kemudian diimplementasikan ke dalam program melalui aktifitas pengkodean. Pengkodean tersebut nantinya dikerjakan menggunakan *tools* Android SDK (*Software Development Kit*) yang memakai bahasa Java sebagai bahasa pemrogramannya. Selain itu, penulis

juga menggunakan layanan Firebase seperti Cloud Firestore sebagai *real-time database* yang mendukung penyimpanan data dengan konsep No-SQL sehingga data yang disinkronkan akan *terupdate* secara otomatis. Selain itu juga penulis menggunakan Firebase Cloud Messaging sebagai layanan yang mendukung *push notification*. Untuk menunjang aktivitas pengkodean ini, penulis menggunakan Android Studio sebagai teks editor dalam menangani semua aktivitas pengkodean dari pengembangan aplikasi.

4. Pengujian Sistem

Pengujian sistem dilakukan untuk mengetahui apakah sistem sudah berjalan dengan baik dengan mengamati fungsional sistem dan mengamati hasil dari eksekusi program tersebut. Pengujian pada penelitian ini nantinya menggunakan metode *blackbox testing* yang mengamati hasil *input* dan *output* dari perangkat lunak tanpa mengetahui struktur kode dari perangkat lunak tersebut. *Blackbox testing* merupakan pengujian yang berfokus pada spesifikasi fungsional dari perangkat lunak, *tester* dapat mendefinisikan kumpulan kondisi *input* dan melakukan pengetesan pada spesifikasi fungsional program (Hidayat dan Putri

2019). Beberapa fungsi dalam sistem yang akan diujikan dapat dilihat dalam tabel 2.3 berikut ini:

Tabel 3.2 Pengujian *Blackbox Testing*

No	Pengujian
1	<i>Login</i> Sistem
2	Daftar Akun Baru
3	Pengisian Form
4	Validasi Form
5	<i>Push</i> Notifikasi

5. Pengoperasian dan Perbaikan

Tahapan terakhir dari metode *waterfall* adalah pengoperasian dan perbaikan. Pada tahapan ini, sistem yang telah melewati pengujian akan mulai dioperasikan oleh pengguna. Perbaikan sistem dilakukan untuk mengantisipasi perubahan, kesalahan ataupun *bug* yang muncul pada aplikasi yang tidak terdeteksi pada saat pengujian.

BAB IV

IMPLEMENTASI DAN PEMBAHASAN

A. Pengumpulan Data

1. Observasi

Selama proses penelitian berlangsung, penulis melakukan observasi atau pengamatan secara langsung di Pondok Pesantren Al-Falah Moga. Pengamatan tersebut meliputi alur pendaftaran yang sudah berjalan, waktu pendaftaran, dan jangkauan calon santri yang akan melakukan pendaftaran.

2. Wawancara

Penulis melakukan wawancara yang bertujuan untuk mengumpulkan data yang akan digunakan sebagai acuan dalam pembuatan aplikasi sistem informasi pendaftaran santri baru. Dalam hal ini, penulis melakukan wawancara dengan narasumber dari pihak Pengurus Pondok Pesantren Al-Falah Moga. Berikut ini merupakan isi dari wawancara beserta jawabannya yang dapat dilihat pada tabel berikut.

Tabel 4.1 Wawancara Penelitian

No	Pertanyaan	Respon
----	------------	--------

1	Bagaimana alur proses pendaftaran santri baru yang sudah berjalan di Pondok Pesantren Al-Falah Moga?	Alur pendaftaran yang berjalan saat ini masih dilakukan secara manual, dimana calon santri datang ke pondok pesantren dan mengisi formulir fisik kemudian melengkapi berkas pendaftaran yang dibutuhkan. Setelah semua berkas diverifikasi, kemudian calon santri akan mengikuti tes seleksi penempatan kelas sesuai dengan jadwal yang telah diberikan bersamaan dengan proses daftar ulang
2	Data dan berkas apa saja yang diperlukan untuk melakukan pendaftaran di Pondok Pesantren Al-Falah Moga?	Data yang dibutuhkan meliputi identitas santri, identitas orangtua/wali santri yang bersangkutan. Sedangkan berkas yang dibutuhkan yaitu scan akta kelahiran, scan kartu keluarga, scan ijazah

		madrasah, dan pas foto dari calon santri
3	Apa saja kendala yang sering dihadapi dalam proses pendaftaran secara manual ini?	Beberapa kendala yang terjadi yaitu kesalahan pengisian formulir, berkas pendaftaran yang hilang, selain itu dibutuhkan waktu dan tenaga yang cukup banyak karena ada beberapa panitia pendaftaran yang memiliki tugas lain di pondok pesantren
4	Apakah ada rencana untuk mengubah sistem pendaftaran ini menjadi berbasis digital?	Ya, pihak pondok sendiri melihat dengan kemajuan teknologi saat ini, digitalisasi proses pendaftaran tentu sangat membantu baik bagi calon santri ataupun pihak pesantren. Pondok pesantren sendiri juga pernah mencoba membuka pendaftaran secara <i>online</i>

		dengan menggunakan <i>google form</i> ketika pandemi
5	Apa yang menjadi pertimbangan jika sistem pendaftaran di pondok pesantren ini diubah menjadi digital	Beberapa hal yang dipertimbangkan yaitu kurangnya sumber daya manusia dalam mengurus pendaftaran secara online. Beberapa pengurus pondok sendiri sudah memiliki tugas ganda sehingga pihak pondok masih mempertimbangkan jika proses pendaftaran beralih ke digital. mungkin itu yang menjadi problem yang harus dipikirkan oleh pihak pondok kedepannya
6	Bagaimana proses pendaftaran yang sudah berjalan setelah mencoba menggunakan	Pada saat pendaftaran menggunakan <i>google form</i> sudah berjalan dengan cukup lancar, beberapa calon santri sudah ada yang mendaftar secara <i>online</i> namun tetap harus ke

	sistem pendaftaran secara <i>online</i> melalui <i>google form</i>?	pondok guna melakukan daftar ulang dan melakukan tes seleksi. Beberapa kendalanya yaitu proses pendaftaran harus melewati beberapa proses dan terbilang kurang efisien karena calon santri harus melakukan komunikasi terlebih dahulu dengan pihak pondok menggunakan platform <i>whatsapp</i>, kemudian <i>link google form</i> baru bisa dikirimkan
--	--	--

B. Implementasi Sistem

Dalam mengembangkan suatu sistem, lingkungan implementasi sangat penting agar aplikasi dapat berfungsi dengan baik. Perancangan aplikasi ini memerlukan komponen pendukung, yaitu perangkat lunak (*software*) dan perangkat keras (*hardware*). Berikut ini merupakan perangkat lunak yang dipakai oleh penulis dalam pembuatan sistem yang dapat dilihat pada tabel berikut.

Tabel 4.2 Perangkat Lunak yang dipakai

No	Perangkat Lunak	Keterangan
1.	Sistem Operasi	Windows 10
2.	<i>Text Editor</i>	Android Studio Jellyfish 2023.3.1
3.	<i>Database</i>	Cloud Firestore
4.	<i>Storage</i>	Cloud Storage Firebase
5.	<i>Push Notification</i>	Firebase Cloud Messaging
6.	Bahasa Pemrograman	Java

Selain perangkat lunak, perangkat keras juga dibutuhkan dalam perancangan sistem. Berikut adalah perangkat keras yang dipakai penulis.

Tabel 4.3 Perangkat Keras yang dipakai

No.	Perangkat Keras	Keterangan
1.	<i>Processor</i>	Intel(R) Core(TM) i7- 1065G7 CPU @ 1.30GHz 1.50 GHz
2.	<i>Hard Disk</i>	512 GB SSD
3.	RAM	16 GB
4.	Monitor	14 inch

a. Implementasi *Database*

Database merupakan kumpulan data yang dikelola sedemikian rupa berdasarkan ketentuan tertentu yang

saling berhubungan sehingga mudah dalam pengelolaannya. Cloud Firestore adalah Layanan Google Cloud Platform untuk penyimpanan data *database NoSQL*. Berbeda dengan *database relasional*, dimana setiap *item* data memiliki hubungan dengan yang lain, *database NoSQL* ini berisi informasi yang tidak terkait langsung dengan yang lain. Ketika data tersimpan di Cloud Firestore, maka data dapat diakses dimanapun dan dengan *stream*, data dapat diperbaharui secara *real-time*. Berikut *database* yang dibuat dalam perancangan sistem.

+ Start collection	+ Add document	+ Start collection
Form >	A8yZGVH7feg8zyVD6BVyJ...	+ Add field
Users	⋮ UEcLDk3VmQV7Nhvp7AE8F... >	Kode Pos: "52354"
	Uc1qxUruIO06HZGFp2mt...	NIK: "123456890"

Gambar 4-1 Database Cloud Firestore

Dalam penggunaan Cloud Firestore terdapat beberapa bagian, berupa *collection*, *document* dan *field*. Cloud Firestore menyimpan data dalam bentuk koleksi (*collection*) dimana *collection* yang dibutuhkan dalam sistem informasi ini adalah *collection* Form untuk menyimpan data pendaftaran dari *user*, dan *collection* Users digunakan untuk menyimpan data akun yang terdaftar ke dalam aplikasi. Dalam *document* terdapat kode unik yang berfungsi mengidentifikasi *user*, kode unik tersebut dibuat otomatis oleh Cloud Firestore

atau dapat ditentukan sendiri oleh admin. *Field* merupakan isi dari *document*, dimana setiap *document* berisi pasangan kunci-nilai (*key-value pairs*) yang merupakan data spesifik untuk setiap pengguna. Sebagai contoh dalam *field* terdapat *key* dengan nama “Kode Pos” dengan *value* berupa “52354” dimana *key* “Kode Pos” ini merupakan salah satu data yang dibutuhkan dalam pengisian form oleh *user*.

b. Hasil Implementasi Sistem

Berikut ini adalah hasil implementasi dari desain antarmuka yang telah dibuat sebelumnya.

1) Halaman Daftar Akun

3:09 PM | 1.2MB/s

< Kembali

Buat Akun
Silahkan daftar terlebih dahulu

Daftar

Username
Calon santri

Nomor Telepon/Whatsapp
08*****

Email
calonsantri@gmail.com

Password

Daftar

Gambar 4-2 Halaman Daftar Akun

Gambar di atas merupakan halaman daftar akun, halaman tersebut digunakan calon santri untuk mendaftar akun baru agar dapat mengakses aplikasi pendaftaran. Data yang dibutuhkan dalam pendaftaran akun yaitu *username*, nomor telepon, alamat *email* dan *password*.

2) Halaman Login Akun



3:25 PM | 0.7KB/s

Selamat Datang
Silahkan masuk terlebih dahulu

Masuk

Email
calonsantri@gmail.com

Password

Masuk

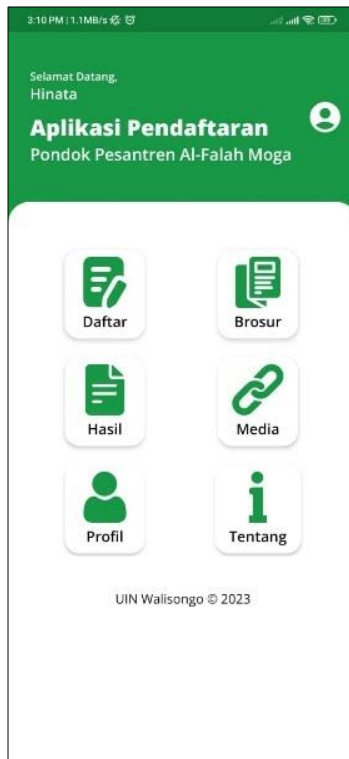
Belum punya akun? Silahkan daftar

Daftar

Gambar 4-3 Halaman Login Akun

Gambar di atas berupa halaman login untuk pengguna, baik itu *user* ataupun admin untuk dapat mengakses halaman sesuai hak aksesnya. Pengguna hanya perlu memasukkan data *email* dan *password* yang telah didaftarkan pada halaman daftar akun.

3) Halaman Beranda User



Gambar 4-4 Halaman Beranda User

Ketika *user* berhasil *login* ke dalam aplikasi, *user* akan diarahkan ke halaman beranda *user*. Halaman tersebut berisi fitur pengisian *form*, brosur pendaftaran, pengumuman hasil pendaftaran, media informasi pondok pesantren, profil akun, dan informasi terkait aplikasi pendaftaran terkait serta kontak penulis.

4) Halaman Form Pendaftaran

Form Pendaftaran

IDENTITAS SANTRI (sesuai kk)

NAMA LENGKAP

TEMPAT LAHIR

TANGGAL LAHIR PILIH

JENIS KELAMIN
☒ Laki-laki ☐ Perempuan

NOMOR INDUK KEPENDUDUKAN (NIK)

ASAL MADRASAH (Jika belum lulus diisi, BELUM LULUS)

PENDIDIKAN FORMAL

ALAMAT (Jalan/Dusun)

RT RW

DESA/KELURAHAN

KECAMATAN

KODE POS

KABUPATEN

IDENTITAS ORANGTUA/WALI (sesuai kk)

PROVINSI

NAMA AYAH

NIK AYAH

PENDIDIKAN FORMAL AYAH

PEKERJAAN AYAH

NAMA IBU

NIK IBU

PENDIDIKAN FORMAL IBU

PEKERJAAN IBU

NOMOR TELEPON ORANGTUA (bisa dihubungi)

UPLOAD SCAN BERKAS

AKTA KELAHIRAN PILIH

KARTU KELUARGA PILIH

IJAZAH MADRASAH PILIH

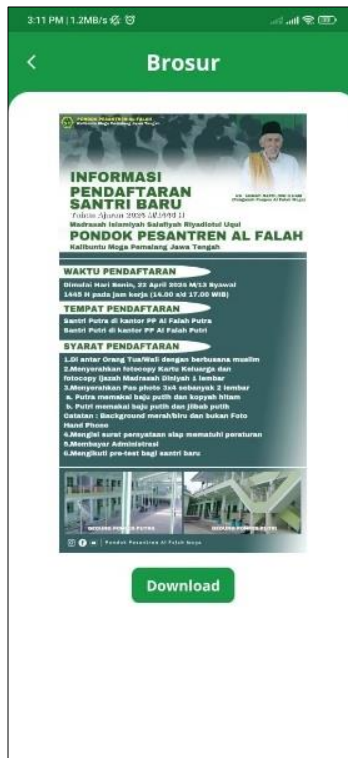
PAS FOTO 3X4 PILIH

Simpan

Gambar 4-5 Halaman Form Pendaftaran

User akan mengisi *form* pendaftaran santri baru yang nantinya akan dibutuhkan dalam seleksi administrasi pondok pesantren. *Form* tersebut berisi data identitas calon santri, identitas orangtua/wali serta berkas yang diperlukan seperti scan akta kelahiran, scan kartu keluarga, scan ijazah, dan scan pas foto.

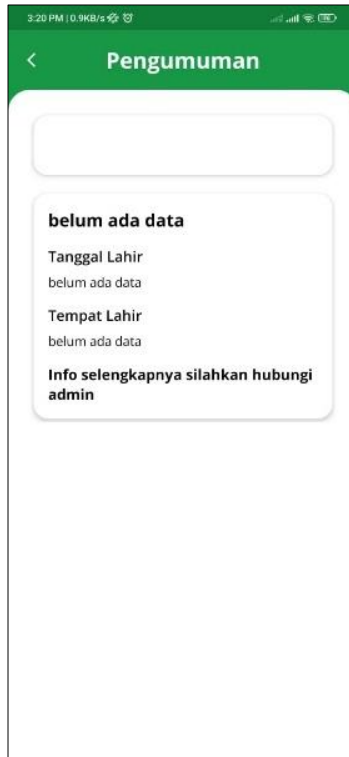
5) Halaman Brosur



Gambar 4-6 Halaman Brosur

Halaman brosur berisi brosur pendaftaran santri baru di Pondok Pesantren Al-Falah Moga mengenai informasi terkait pendaftaran baik waktu, tempat ataupun syarat yang dibutuhkan. *User* dapat menyimpan brosur tersebut dengan menekan tombol *download* dan secara otomatis tersimpan ke penyimpanan telepon *user*.

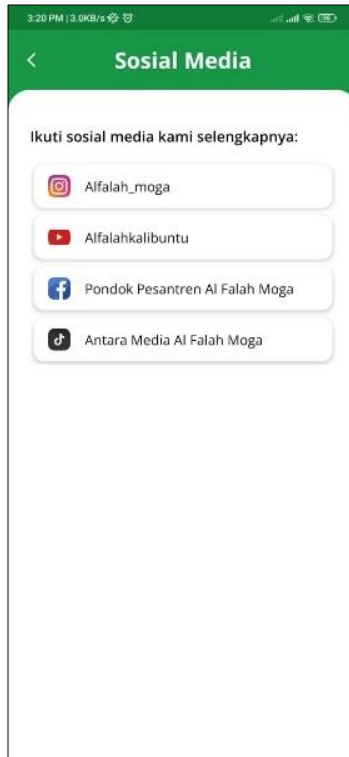
6) Halaman Pengumuman



Gambar 4-7 Halaman Pengumuman

Ketika *form* pendaftaran berhasil di validasi, *user* akan mengetahui apakah berkasnya lolos seleksi administrasi ataupun tidak dengan melihat halaman pengumuman. Jika *user* belum mengisi *form* pendaftaran, halaman tersebut tidak akan menampilkan data.

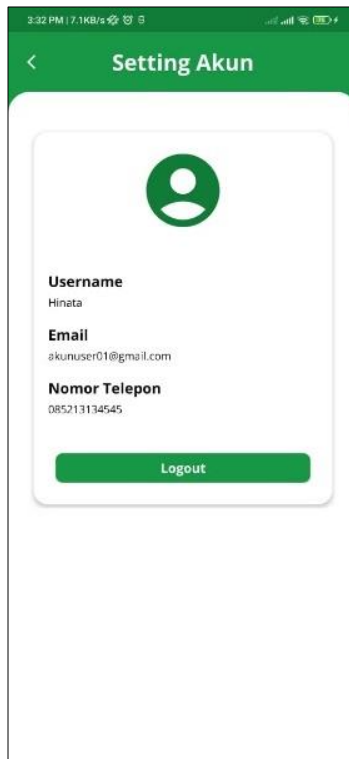
7) Halaman Media



Gambar 4-8 Halaman Media

Halaman media berisi informasi kontak Pondok Pesantren Al-Falah Moga berupa *link* sosial media yang dapat diakses oleh calon santri untuk mengetahui informasi lebih lanjut tentang Pondok Pesantren Al-Falah Moga.

8) Halaman Profil User



Gambar 4-9 Halaman Profil User

Halaman profil *user* berisi informasi dari akun *user* yang telah didaftarkan. Beberapa informasi yang ditampilkan yaitu *username*, *email* dan nomor telepon dari *user*. Terdapat tombol *logout* untuk mengakhiri sesi dari *user*.

9) Halaman Info Aplikasi



Gambar 4-10 Halaman Info Aplikasi

Halaman info aplikasi merupakan halaman tentang penjelasan singkat mengenai aplikasi pendaftaran santri baru Pondok Pesantren Al-Falah Moga dan informasi kontak dari penulis.

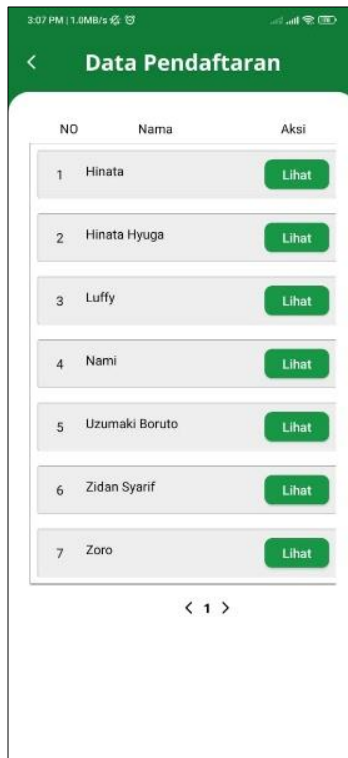
10) Halaman Beranda Admin



Gambar 4-11 Halaman Beranda Admin

Ketika admin berhasil *login*, aplikasi akan menampilkan beranda admin. Halaman tersebut berisi data pendaftaran dari calon santri, fitur *update* brosur, fitur verifikasi pendaftaran, dan profil akun.

11) Halaman Data Pendaftaran Admin



Gambar 4-12 Halaman Data Pendaftaran

Halaman data pendaftaran berisi nama calon santri yang telah mengisi *form* pendaftaran. Admin dapat melihat *form* yang telah diisi oleh calon santri dengan menekan tombol lihat.

12) Halaman Lihat Data Pendaftaran





Gambar 4-13 Halaman Lihat Data Pendaftaran

Setelah admin menekan tombol lihat, maka aplikasi akan menampilkan data *form* pendaftaran yang telah diisi oleh *user*. Admin dapat memvalidasi berkas tersebut apakah sudah memenuhi administrasi atau tidak dengan menekan tombol terima atau tidak diterima.

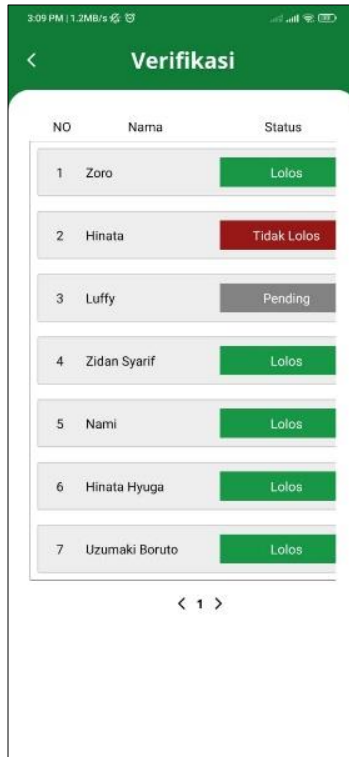
13) Halaman Edit Brosur



Gambar 4-14 Halaman Edit Brosur

Halaman edit brosur dapat digunakan oleh admin untuk *mengupdate* brosur menyesuaikan dengan tahun penerimaan santri baru di Pondok Pesantren Al-Falah Moga.

14) Halaman Verifikasi



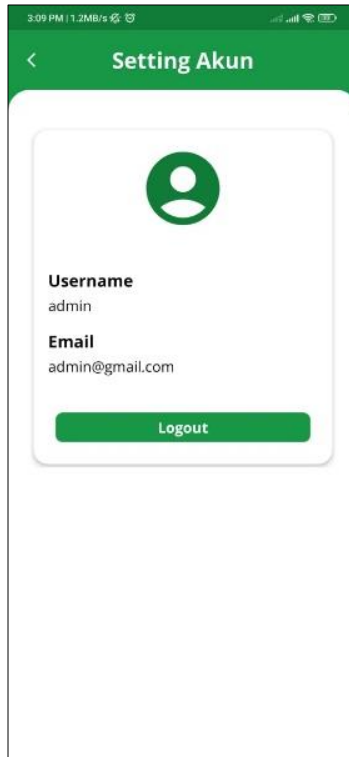
NO	Nama	Status
1	Zoro	Lolos
2	Hinata	Tidak Lolos
3	Luffy	Pending
4	Zidan Syarif	Lolos
5	Nami	Lolos
6	Hinata Hyuga	Lolos
7	Uzumaki Boruto	Lolos

< 1 >

Gambar 4-15 Halaman Verifikasi

Halaman verifikasi menampilkan nama *user* yang sudah ataupun belum divalidasi oleh admin dalam seleksi administrasi. Jika berkas telah divalidasi maka akan menampilkan status lolos ataupun tidak lolos sesuai dengan hasil berkas yang telah divalidasi oleh admin. Jika belum divalidasi, maka akan menampilkan status *pending*.

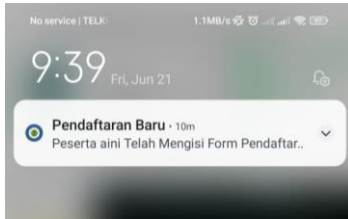
15) Halaman Profil Admin



Gambar 4-16 Halaman Profil Admin

Admin dapat mengetahui tentang informasi akun pada halaman profil admin. Halaman tersebut juga terdapat tombol *logout* untuk mengakhiri sesi akun dari admin dan mengarahkan aplikasi ke halaman *login*.

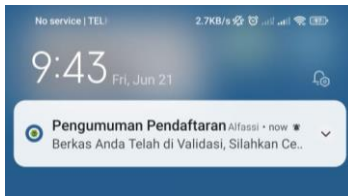
16) Notifikasi Pendaftaran Baru



Gambar 4-17 Notifikasi Pendaftaran Baru

Notifikasi pendaftaran santri baru akan tampil pada aplikasi admin ketika ada calon santri baru yang telah mengisi *form* pendaftaran.

17) Notifikasi Saat Validasi Pendaftaran



Gambar 4-18 Notifikasi Saat Validasi Pendaftaran

Notifikasi ini akan tampil ketika admin berhasil memvalidasi data pendaftaran yang telah diisi oleh calon santri secara akurat sesuai dengan data yang telah divalidasi.

c. Kode Pemrograman

1) Proses *Login*

```
private void masuk(String email, String
password) {
    progressDialog.show();
    mAuth.signInWithEmailDanPassword(email,
password).addOnSuccessListener(new
OnSuccessListener<AuthResult>() {
    @Override
    public void onSuccess(AuthResult
authResult) {
        cekRole(authResult.getUser().getUid());
        progressDialog.dismiss();
    }
}).addOnFailureListener(new
OnFailureListener() {
    @Override
    public void onFailure(@NonNull Exception
e) {
        progressDialog.dismiss();
        Toast.makeText(getApplicationContext(),
        "Login gagal!",
        Toast.LENGTH_SHORT).show();
    }
});
}
```

Dalam proses *login* ada 2 parameter yang dibutuhkan yaitu *email* dan *password*. Proses dimulai dengan *progressDialog* yang bertujuan bahwa proses *login* sedang berlangsung. Kemudian metode *signInWithEmailDanPassword* dari objek *mAuth* dipanggil. Metode ini merupakan bagian dari Firebase Authentication dan berfungsi

memverifikasi kredensial pengguna terhadap data yang disimpan di Cloud Firestore. Jika autentikasi berhasil kemudian menjalankan metode *cekRole* dimana dalam metode ini bertujuan untuk memeriksa peran pengguna (*role*) yang menentukan hak akses terhadap aplikasi. Kemudian jika autentikasi gagal maka akan menampilkan pesan *login* gagal.

```
private void cekRole(String uid) {
    DocumentReference df =
    firestore.collection("Users").document(uid);
    df.get().addOnSuccessListener(new
    OnSuccessListener<DocumentSnapshot>() {
        @Override
        public void onSuccess(DocumentSnapshot
        documentSnapshot) {
            //identifikasi role

            if (documentSnapshot.getString("isAdmin")
            != null) {
                startActivity(new
                Intent(getApplicationContext(),
                BerandaAdminActivity.class));
            }
            if (documentSnapshot.getString("isUser")
            != null) {
                startActivity(new
                Intent(getApplicationContext(),
                BerandaUserActivity.class));
            }
        }
    });
}
```

Setelah autentikasi berhasil maka akan memanggil metode *cekRole*, dimana proses dimulai dengan mengambil UID (*User ID*) pengguna yang telah berhasil masuk. UID ini kemudian digunakan untuk membuat referensi dokumen dalam koleksi "Users" di Cloud Firestore. Setelah referensi dokumen dibuat, metode *cekRole* mengambil data dokumen secara asinkron menggunakan metode *get*. Jika dokumen memiliki *field* "*isAdmin*" yang tidak berupa *null*, maka pengguna diidentifikasi sebagai admin dan mengarahkan pengguna tersebut ke halaman beranda admin. Sebaliknya, jika dokumen memiliki *field* "*isUser*" yang tidak berupa *null*, pengguna diidentifikasi sebagai *user* biasa dan diarahkan ke halaman beranda *user*.

2) Proses Daftar Akun

```

private void Daftar(String
username,String nomor, String email,
String password) {
progressDialog.show();
 mAuth.createUserWithEmailDanPassword(email,
password).addOnCompleteListener(new
OnCompleteListener<AuthResult>() {
    @Override
    public void onComplete(@NonNull
Task<AuthResult> task) {
        if (task.isSuccessful() &&
task.getResult() != null) {
            FirebaseUser firebaseUser =
task.getResult().getUser();
            DocumentReference df =
fStore.collection("Users").document(firebaseUser.getUid());
            Map<String, Object> userInfo = new
HashMap<>();
            userInfo.put("Username",
username.toString());
            userInfo.put("Email", email.toString());
            userInfo.put("Nomor", nomor.toString());
            userInfo.put("isUser", "1");
            df.set(userInfo);
        } else {
            Toast.makeText(getApplicationContext(),
task.getException().getLocalizedMessage(),
Toast.LENGTH_SHORT).show();
        }
        progressDialog.dismiss();
    }
});
}

```

Metode Daftar ini berfungsi untuk mendaftarkan pengguna baru ke dalam aplikasi

menggunakan Firebase Authentication dan Cloud Firestore sebagai penyimpanan data pengguna. Metode ini menerima empat parameter: *username*, nomor, *email*, dan *password*. Proses dimulai dengan menampilkan *progressDialog* untuk memberi tahu pengguna bahwa proses pendaftaran sedang berlangsung. Selanjutnya, metode *createUserWithEmailDanPassword* dari objek *mAuth* (Firebase Authentication) dipanggil untuk membuat akun baru menggunakan *email* dan *password* yang dilakukan secara asinkron. Jika pendaftaran berhasil maka akan membuat referensi dokumen dalam koleksi "Users" di Cloud Firestore menggunakan UID dari *user* yang telah dibuat. *UserInfo* dibuat untuk menyimpan informasi pengguna, termasuk *username*, *email*, dan nomor. Selain itu, *field* tambahan "*isUser*" dengan nilai "1" ditambahkan untuk menandai bahwa pengguna ini adalah pengguna biasa. Referensi dokumen diatur dengan data dari *userInfo* menggunakan metode *set*.

3) Proses Pengiriman Form

```
private void uploadData() {
    FirebaseFirestore fStore =
    FirebaseFirestore.getInstance();

    String NamaLengkap =
    etNama.getText().toString();
    String TempatLahir =
    etTempatlahir.getText().toString();

    Map<String, Object> user = new
    HashMap<>();
    user.put("Nama Lengkap", NamaLengkap);
    user.put("Tempat Lahir", TempatLahir);
}
```

Dalam metode *uploadData* proses pertama yang dilakukan adalah dengan membuat variabel *fStore* untuk menyimpan *instance* dari Cloud Firestore kemudian mendeklarasikan variabel sebagai contoh *NamaLengkap* dengan tipe data String yang diambil dari hasil *input user* ke dalam *editText* saat mengisi form. Setelah membuat String selanjutnya akan disimpan ke dalam map *user* dengan menggunakan metode *put* untuk menambahkan pasangan *key-value*, dalam hal ini *key* yang dimaksud adalah “Nama Lengkap” dengan *value* *NamaLengkap* yang diambil dari *edittext* yang telah diisi oleh *user* saat mengisi form.

```

fStore.collection("Form").document(fUser.
getUid()).set(user)

if (userSnapshot.exists()) {

String tokenFCM =
userSnapshot.getString("token");
FcmNotificationSender
fcmNotificationsSender = new
FcmNotificationSender(tokenFCM,
"Pendaftaran Baru", "Peserta "+
NamaLengkap + " Telah Mengisi Form
Pendaftaran", this);
fcmNotificationsSender.sendNotification()
;

// Pastikan untuk memanggil metode untuk
mengirim notifikasi

Toast.makeText(getApplicationContext(),
"Berhasil simpan data!",
Toast.LENGTH_SHORT).show();
Intent intent = new Intent(this,
BerandaUserActivity.class);
startActivity(intent);
finish();

.addOnFailureListener(e -> {
Toast.makeText(getApplicationContext(),
"Gagal menyimpan data",
Toast.LENGTH_SHORT).show();
progressDialog.dismiss();
});
}

```

Setelah disimpan melalui map *user*, kemudian mengakses koleksi Form dalam Cloud Firestore dan menambahkan sebuah *document* baru yang diidentifikasi menggunakan UID *user* yang akan

mengirim data. Jika berhasil, kemudian membuat variabel baru dengan tipe data String yang mengambil *field* "token" yang berisi token FCM admin. Token ini nantinya menjadi tujuan pengiriman notifikasi. Jika data sudah berhasil disimpan dan notifikasi berhasil dikirimkan, maka akan menampilkan pesan berhasil dan mengarahkan *user* kembali ke halaman beranda. Dan jika gagal menyimpan data, maka akan menampilkan peringatan gagal.

4) Proses Validasi

```
DocumentReference docRef =  
fStore.collection("Form").document(documentId);  
DocumentReference userRef =  
fStore.collection("Users").document(documentId);
```

Kode diatas merupakan kode untuk membuat referensi dokumen berdasarkan *documentId* dalam koleksi Form untuk menambahkan *field* baru berupa status serta dalam koleksi Users untuk mengambil token FCM sebagai tujuan pengiriman notifikasi.

```

userRef.get().addOnSuccessListener(userSnapshot -> {
    if (userSnapshot.exists()) {
        String tokenFCM =
            userSnapshot.getString("token");

        docRef.update("Status", status)
            .addOnSuccessListener(aVoid -> {

                FcmNotificationSender
                fcmNotificationsSender = new
                FcmNotificationSender(tokenFCM,
                    "Pengumuman Pendaftaran", "Berkas Anda
                    Telah di Validasi, Silahkan Cek!!",
                    this);
                fcmNotificationsSender.sendNotification()
                ;

                Toast.makeText(getApplicationContext(),
                    "Status diperbarui " + status,
                    Toast.LENGTH_SHORT).show();
                Intent intent = new
                Intent(LihatDataAdminActivity.this,
                    VerifikasiAdminActivity.class);
                startActivity(intent);
                progressDialog.dismiss();
                finish();
            }
        );
    }
});

```

Mengambil data pengguna dari koleksi Users, data yang akan diambil yaitu token FCM yang berfungsi sebagai objek pengiriman notifikasi. Setelah mengambil data, kemudian menambah status *user* di dalam koleksi Form. Jika status telah ditambahkan, sistem akan menjalankan metode untuk mengirimkan notifikasi ke *user* terkait, dan menampilkan pesan status telah berhasil

diperbarui dan mengarahkan admin ke halaman verifikasi. Jika gagal memperbarui status dan mengambil token *user*, sistem akan menampilkan pesan peringatan.

5) Proses Pengiriman Notifikasi

```
public void sendNotification () {
    RequestQueue requestQueue =
    Volley.newRequestQueue(context);
    JSONObject mainObj = new JSONObject();
    try {
        JSONObject messageObject = new
        JSONObject();
        JSONObject notificationObject = new
        JSONObject();

        notificationObject.put("title", title);
        notificationObject.put("body", body);
        messageObject.put("token", userFcmToken);
        messageObject.put("notification",
        notificationObject);
        mainObj.put("message", messageObject);

        JSONObjectRequest request = new
        JSONObjectRequest(Request.Method.POST,
        postUrl, mainObj, response -> {
            Log.d("FCM", "Notification sent
            successfully: " + response.toString());
        }, volleyError -> {
            Log.e("FCM", "Error sending notification:
            " + volleyError.toString());
        })
    }
```

Proses dimulai dengan pembuatan antrean permintaan jaringan menggunakan *Volley* (*RequestQueue requestQueue*). Ini memungkinkan

aplikasi untuk mengelola dan menjalankan permintaan HTTP secara efisien. Selanjutnya, objek JSON utama (*mainObj*) dibuat untuk menyimpan data notifikasi. Di dalam objek ini, dibuat dua objek JSON tambahan: *messageObject* dan *notificationObject*. *NotificationObject* berisi judul dan isi notifikasi, sementara *messageObject* diisi dengan token FCM pengguna (*userFcmToken*) dan pesan notifikasi yang akan dikirimkan. Kemudian, permintaan JSON POST (*JsonObjectRequest request*) dibuat untuk mengirim data ke URL server FCM (*postUrl*). Permintaan ini mengandung objek JSON utama (*mainObj*) yang telah dibuat. Dalam permintaan ini, ditentukan metode POST dan URL tujuan. Respon sukses dan kesalahan diatur dengan menampilkan *log* yang sesuai.

6) Proses Penerimaan Notifikasi

```
public void
onMessageReceived(RemoteMessage
remoteMessage) {
    if (remoteMessage.getNotification() !=
        null) {
        sendNotification(remoteMessage.getNotific
            ation().getTitle(),
            remoteMessage.getNotification().getBody()
        );
    }
}
```

Ketika aplikasi menerima pesan dari Firebase Cloud Messaging (FCM), metode *onMessageReceived* diaktifkan untuk memproses pesan tersebut. Jika pesan berisi notifikasi, metode ini mencatat isi notifikasi dan kemudian memanggil *sendNotification* untuk menampilkan notifikasi lokal kepada pengguna dengan judul dan isi pesan yang diterima. Proses ini memastikan bahwa pengguna diberitahu tentang pesan penting yang dikirim melalui FCM dan memberikan informasi yang relevan secara *real-time* melalui notifikasi lokal di perangkat mereka.

7) Proses *Logout*

```
logout_admin.setOnClickListener(v -> {
    FirebaseAuth.getInstance().signOut();
    startActivity(new
    Intent(getApplicationContext(),
    LoginActivity.class));
    finish();
});
```

Ketika tombol *logout* diklik, pengguna keluar dengan memanggil metode *signOut* dari Firebase Authentication, yang menghapus status autentikasi mereka. Selanjutnya, aplikasi mengarahkan pengguna kembali ke halaman *login* dengan menggunakan *intent*. Kemudian ditutup dengan memanggil *finish*, memastikan bahwa pengguna

tidak bisa kembali ke halaman *logout* melalui tombol kembali pada perangkat mereka. Proses ini membantu menjaga keamanan dan manajemen sesi pengguna dalam aplikasi.

C. Pengujian Sistem

Pada tahap pengujian ini, peneliti menggunakan metode untuk mengetahui apakah sistem sudah berjalan dengan baik dengan mengamati fungsionalitas sistem tanpa memeriksa struktur kode dari sistem tersebut. Berikut hasil skenario pengujian pada penelitian ini.

Tabel 4.4 Pengujian *Blackbox Login*

Nama Fitur	Detail Test Case	Hasil yang Diharapkan	Hasil Pengujian
Login	User login mengisi semua kolom dengan benar	Login berhasil dan aplikasi menampilkan halaman beranda sesuai hak akses	Berhasil
	User login hanya mengisi salah satu kolom	Menampilkan Error, dan user harus mengisi kolom	Berhasil

	User login tanpa mengisi kolom	Menampilkan Error, dan user harus mengisi kolom	Berhasil
	User login mengisi semua kolom dengan salah	Menampilkan Error, user harus mengisi dengan akun yang sudah terdaftar	Berhasil

Tabel 4.5 Pengujian *Blackbox* Daftar

Nama Fitur	Detail Test Case	Hasil yang Diharapkan	Hasil Pengujian
Daftar	User daftar mengisi semua kolom dengan benar	Daftar berhasil, dan mengarahkan ke halaman user	Berhasil
	User daftar hanya mengisi salah satu kolom	Menampilkan Error, dan user harus mengisi kolom	Berhasil
	User daftar tanpa mengisi kolom	Menampilkan Error, dan user harus mengisi kolom	Berhasil

Tabel 4.6 Pengujian *Blackbox Form* Daftar

Nama Fitur	Detail Test Case	Hasil yang Diharapkan	Hasil Pengujian
Form Daftar	User mengisi form dan mengupload gambar dengan benar	Data tersimpan dan berhasil mengirim notifikasi ke admin	Berhasil
	User mengisi form tanpa mengupload gambar	Menampilkan Error, dan user harus memilih gambar	Berhasil
	User mengupload gambar tanpa mengisi form	Menampilkan Error, dan user harus mengisi form	Berhasil
	User tidak mengisi form dan tidak mengupload gambar	Menampilkan Error, user harus mengisi form dan mengupload gambar	Berhasil

Tabel 4.7 Pengujian *Blackbox Validasi* Pendaftaran

Nama Fitur	Detail Test Case	Hasil yang Diharapkan	Hasil Pengujian
------------	------------------	-----------------------	-----------------

Validasi pendaftaran	Admin validasi user dengan tombol lolos	Admin berhasil validasi user, mengirim notifikasi dan menampilkan halaman validasi	Berhasil
	Admin validasi user dengan tombol tidak lolos	Admin berhasil validasi user, mengirim notifikasi dan menampilkan halaman validasi	Berhasil

Tabel 4.8 Pengujian *Blackbox Push* Notifikasi

Nama Fitur	Detail Test Case	Hasil yang Diharapkan	Hasil Pengujian
Push Notifikasi	User berhasil mengirim notifikasi ke admin saat berhasil mengisi form	Tampil notifikasi pendaftaran baru di aplikasi admin	Berhasil
	Admin mengirim notifikasi ke user yang	Tampil notifikasi pengumuman pendaftaran	Berhasil

	dituju saat berhasil validasi form	di aplikasi user	
--	--	---------------------	--

BAB V

KESIMPULAN DAN SARAN

A. Kesimpulan

Berdasarkan hasil penelitian dengan judul “Sistem Informasi Pendaftaran Santri Baru Berbasis *Mobile* Dengan *Push Notification* Menggunakan Firebase Cloud Messaging (FCM) di Pondok Pesantren Al-Falah Moga”, maka dapat disimpulkan sebagai berikut:

1. Sistem Informasi Pendaftaran Santri Baru di Pondok Pesantren Al-Falah Moga telah berhasil dirancang, dengan penambahan fitur *push notification* menggunakan FCM.
2. Fitur *push notification* menggunakan Firebase Cloud Messaging (FCM) dapat berjalan dengan baik dan sesuai dengan yang diharapkan.
3. Berdasarkan hasil pengujian *black box testing* yang telah dilakukan, sistem informasi telah berjalan dengan baik dan dapat menjalankan fungsi pendaftaran santri baru.

B. Saran

Dari hasil penelitian yang telah dilaksanakan, tentunya masih banyak kekurangan dan perbaikan baik dalam sistem ataupun tampilan aplikasi yang

sudah dibuat. Oleh karena itu, penulis memberikan beberapa saran sebagai berikut:

1. Peningkatan sistem dan proses *debugging* yang lebih lanjut untuk meningkatkan kualitas aplikasi kedepannya.
2. Meningkatkan kualitas tampilan yang telah dibuat, agar lebih menarik dan lebih mudah digunakan (*user friendly*).

DAFTAR PUSTAKA

- Aceng Abdul Wahid. 2020. "Analisis Metode Waterfall Untuk Pengembangan Sistem Informasi." *Jurnal Ilmu-Ilmu Informatika Dan Manajemen STMIK*, no. November, 1–5.
- Aliman, Wilianti. 2021. "PERANCANGAN PERANGKAT LUNAK UNTUK MENGGAMBAR DIAGRAM BERBASIS ANDROID" 6 (6): 6.
- Anggraini, Yeni, Donaya Pasha, dan Aan Setiawan. 2020. "Sistem Informasi Penjualan Sepeda Berbasis Web Menggunakan Framework Codeigniter (Studi Kasus: Orbit Station)." *Jurnal Teknologi Dan Sistem Informasi (JTSI)* 1 (2): 64–70.
- Asmara, Jimi. 2019. "Rancang Bangun Sistem Informasi Desa Berbasis Website (Studi Kasus Desa Netpala)." *Jurnal Pendidikan Teknologi Informasi (JUKANTI)* 2 (1): 1–7.
- Auzaieafif19. n.d. "Metode Pengembangan Model Waterfall." <https://auzaieafif19.wordpress.com/rekayasa-perangkat-lunak/metode-pengembangan-model-waterfall/>.
- Cahyadi, Rudy, Ariesta Damayanti, Irwan Setiawan, Teknologi Rekayasa Multimedia Poltek Negeri Media Kreatif Jakarta Jl Srengseng Sawah, Jakarta Selatan, Teknik Informatika STMIK AKAKOM Jl Raya Janti, dan Karangjambé Yogyakarta. 2019. "Teknologi Firebase Untuk Aplikasi Lapor Akakom." *Jurnal Informatika Dan Komputer* 4 (1): 11–17.
- Danuri, Muhamad. 2019. "Development dan Transformation of Digital Technology." *Infokam XV (II)*: 116–23.

- Darmawan, Rifqi. 2019. "IMPLEMENTASI PUSH NOTIFICATION SCHEDULER PADA FIREBASE CLOUD MESSAGING UNTUK APLIKASI PENGINGAT KEGIATAN ORGANISASI MENGGUNAKAN FLUTTER BERBASIS MOBILE." *Proceedings of the Institution of Mechanical Engineers, Part J: Journal of Engineering Tribology* 224 (11): 122–30.
- Darnita, Yulia, dan Muntahanah Muntahanah. 2021. "Rancang Bangun Aplikasi Mobile Penjadwal Perkuliahan Dengan Firebase Dengan Realtime Notification." *Pseudocode* 8 (1): 58–65. <https://doi.org/10.33369/pseudocode.8.1.58-65>.
- Faisol, Ahmad, dan Febi Rahmadianto. 2018. "Realtime Notification Pada Aplikasi Berbasis Web." *Mnemonic* 1 (2): 14–17.
- Fernando, Ferry. 2020. "Perancangan User Interface (Ui) & User Experience (Ux) Aplikasi Pencari Indekost Di Kota Padangpanjang." *TANRA: Jurnal Desain Komunikasi Visual Fakultas Seni Dan Desain Universitas Negeri Makassar* 7 (2): 101. <https://doi.org/10.26858/tanra.v7i2.13670>.
- Firebase. n.d. "Firebase Cloud Messaging." <https://firebase.google.com/docs/cloud-messaging?hl=id>.
- Fitri, Riskal, dan Syarifuddin Ondeng. 2022. "Pesantren Di Indonesia: Lembaga Pembentukan Karakter." *Jurnal Al-Urwatul Wutsqa: Kajian Pendidikan Islam* 2 (1): 42–54.
- Hidayat, Taufik, dan Hendar Dini Putri. 2019. "Pengujian Portal Mahasiswa Pada Sistem Informasi Akademik (SINA) Menggunakan Black Box Testing Dengan Metode Equivalence Partitioning Dan Boundary Value Analysis." *Jutis* 7 (1): 83–92.

- Kahfi, Shofiyullahul, dan Ria Kasanova. 2020. "Manajemen Pondok Pesantren Di Masa Pandemi Covid-19 (Studi Pondok Pesantren Mambaul Ulum Kedungadem Bojonegoro)." *Jurnal Pendidikan Berkarakter* 3 (1): 26–30.
- Krisdiyanto, Gatot, Muflikha Muflikha, Elly Elvina Sahara, dan Choirul Mahfud. 2019. "Sistem Pendidikan Pesantren Dan Tantangan Modernitas." *Tarbawi : Jurnal Ilmu Pendidikan* 15 (1): 11–21.
<https://doi.org/10.32939/tarbawi.v15i1.337>.
- Laia, Firdaus. 2023. "Perancangan Aplikasi Android Untuk Mengoptimalkan Proses Pesanan Pada Restoran Dengan Metode Prototipe" 7:1482–90.
- Malik, I, dan H Zakaria. 2023. "Implementasi Sistem Deteksi Banjir Menggunakan Mikrokontroler Nodemcu Esp8266 Melalui Notifikasi Whatsapp Gateway Dan" *LOGIC: Jurnal Ilmu Komputer Dan ...* 1 (4): 746–52.
- Material. n.d. "Android Notifications." <https://m2.material.io/design/platform-guidance/android-notifications.html>.
- Meidina, Mita, Junadhi Junadhi, Unang Rio, dan Yoyon Efendi. 2022. "Penerapan Fitur Firebase Cloud Messaging Pada Sistem Kontrol Pembayaran Iuran BPJS Ketenagakerjaan Berbasis Mobile." *Building of Informatics, Technology dan Science (BITS)* 4 (1): 137–48.
<https://doi.org/10.47065/bits.v4i1.1603>.
- Nur Anisa, Fidha, dan Setyoningsih Wibowo. 2022. "Penggunaan Machine Learning Dalam Aplikasi Rekomendasi Tempat Wisata Jawa Tengah (Se-Jawat Tourish)." *Science And Engineering National Seminar* 7 (7).

- Nurzam, Fariz D, Ika Nur Fajri, dan Donni Prabowo. 2017. "Rancang Bangun Aplikasi Media Laporan Aspirasi Dengan Firebase Cloud Messaging Berbasis Mobile." *Semnasteknomedia Online* 5 (1): 4-5-37.
- Purwanto, Hari Lugis. 2023. "Penerapan Push Notification Pada Sistem Monitoring Belajar Berbasis Web Menggunakan Firebase Cloud Messaging." *JSR: Jaringan Sistem Informasi Robotik* 7 (1): 13-20. <https://doi.org/10.58486/jsr.v7i1.203>.
- Ridwan, Muhammad, Iskandar Fitri, dan Benrahman Benrahman. 2021. "Rancang Bangun Marketplace Berbasis Website Menggunakan Metodologi Systems Development Life Cycle (SDLC) Dengan Model Waterfall." *Jurnal JTIK (Jurnal Teknologi Informasi Dan Komunikasi)* 5 (2): 173. <https://doi.org/10.35870/jtik.v5i2.209>.
- Sa'diyah, Siti Khalimatus. 2019. "Upaya Ustadz Dalam Dengan Meningkatkan Kemahiran Nahwu Santri Mengimplementasikan Metode Hafalan Di Pondok Pesantren Al- Falah Moga Pemalang," 40.
- Somya, Ramos, dan Monika Aprillia. 2019. "Perancangan Aplikasi Push Notification Center Dengan Teknologi Firebase Cloud Messaging Di Pt. Sumber Trijaya Lestari." *Simetris: Jurnal Teknik Mesin, Elektro Dan Ilmu Komputer* 10 (1): 211-22. <https://doi.org/10.24176/simet.v10i1.2935>.
- Syarif, Muhamad, dan Wahyu Nugraha. 2020. "Pemodelan Diagram UML Sistem Pembayaran Tunai Pada Transaksi E-Commerce." *Jurnal Teknik Informatika Kaputama (JTIK)* 4 (1): 70 halaman.

- Tommy, Lukas, Delpiah Wahyuningsih, dan Parlia Romadiana. 2020. "Pengembangan Aplikasi Penerimaan Mahasiswa Baru Berbasis Android Dengan Push Notification Di STMIK Atma Luhur." *Jurnal Sisfokom (Sistem Informasi Dan Komputer)* 9 (1): 108–21. <https://doi.org/10.32736/sisfokom.v9i1.813>.
- Triawan, Anggra, dan Andika Ramot Yesaya Siboro. 2021. "Penerapan Application Programming Interface (API) Pada Push Notification Untuk Informasi Monitoring Stok Barang Minim." *Teknois: Jurnal Ilmiah Teknologi Informasi Dan Sains* 11 (2): 107–14. <https://doi.org/10.36350/jbs.v11i2.120>.
- Yulianeu, Aneu, dan Muhamad Sodik. 2019. "Perancangan Sistem Informasi Penerimaan Santri Baru Di Pondok Pesantren Sukamanah." *Jurnal Manajemen Informatika* 6 (2): 21–30.

LAMPIRAN

Lampiran 1 : Lembar Persetujuan Pembimbing

HALAMAN PERSETUJUAN

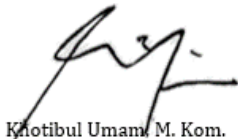
Proposal Skripsi ini telah disetujui oleh Pembimbing untuk dilaksanakan.

Disetujui pada:

Hari : Jum'at

Tanggal : 10 November 2023

Pembimbing I.



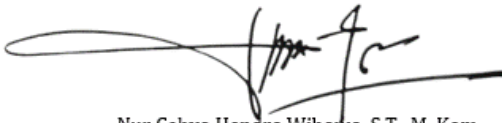
Khotibul Umam, M. Kom.
NIP. 197908272011011007

Pembimbing II.



Siti Nur Afni, M. Kom.
NIP. 198401312018012001

Mengetahui
Kepala Jurusan Teknologi Informasi



Nur Cahyo Hendro Wibowo, S.T., M. Kom.
NIP. 197312222006041001

Lampiran 2 : Lembar Pengesahan Proposal



KEMENTERIAN AGAMA REPUBLIK INDONESIA
UNIVERSITAS ISLAM NEGERI WALISONGO SEMARANG
FAKULTAS SAINS DAN TEKNOLOGI
Jl.Prof.Dr.Hamka Km.1 Semarang Telp. 024 76433366 Semarang 50185
E-mail: fst@walisongo.ac.id. Web : <http://fst.walisongo.ac.id>

PENGESAHAN

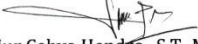
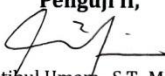
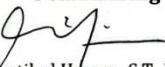
Naskah proposal berikut ini:

Judul : Sistem Informasi Pendaftaran Santri Baru Berbasis
Mobile dengan Push Notification Menggunakan
Firebase Cloud Messaging (FCM) di Pondok
Pesantren Al-Falah Moga
Penulis : Muhammad Dhiyaul Allam
NIM : 1908096047
Jurusan : Teknologi Informasi

Telah diujikan dalam ujian komprehensif oleh Dewan Penguji
Fakultas Sains dan Teknologi UIN Walisongo dan dapat
diterima sebagai salah satu syarat memperoleh gelar sarjana
dalam Teknologi Informasi.

Semarang, 3 Juni 2024

DEWAN PENGUJI

Penguji I,  Nur Cahyo Hendro., S.T., M.Kom. NIP.197312222006041001	Penguji II,  Dr. Khotibul Umam., S.T., M.Kom. NIP.197908272011011007
Penguji III,  Hery Mustofa., M.Kom. NIP.198703172019031007	Penguji IV,  Mokhamad Ikhlil Mustofa, M.Kom. NIP. 198808072019031010
Pembimbing I,  Dr. Khotibul Umam., S.T., M.Kom. NIP. 197908272011011007	Pembimbing II,  Siti Nur'aini, M.Kom. NIP.199107032019031006

Lampiran 3 : Dokumentasi Pengujian *Blackbox Testing*



Lampiran 4 : Formulir Pernyataan Pendaftaran



MADRASAH ISLAMIYAH SALAFIYAH RIYADLOTUL UQUL
PONDOK PESANTREN AL FALAH
KALIBUNTU MOGA PEMALANG



SURAT PERNYATAAN

Yang bertanda tangan di bawah ini :

Nama _____

No. HP. _____

Alamat : Jalan.....

Desa RT : RW :

Kecamatan

Kabupaten

Wali dari

Tempat,Tgl Lahir :

Asal Madrasah :

Sekolah Formal :

Menyerahkan dengan sepenuhnya putra putri kami untuk menjadi santri di Madrasah Islamiyah Salafiyah Riyadlotul Uqul Pondok Pesantren Al Falah Kalibuntu Moga serta siap mematuhi semua peraturan dan sanksi yang berlaku baik yang tertulis maupun yang tidak tertulis.

Demikian surat pernyataan ini kami buat dengan penuh kesadaran dan untuk digunakan sebagaimana mestinya.

Moga,

Wali Santri,

Santri,

Pengurus,

.....

Lampiran 5 : Daftar Riwayat Hidup

RIWAYAT HIDUP**A. Identitas Diri**

1. Nama Lengkap : Muhammad Dhiyaul Allam
2. Tempat & Tanggal Lahir : Pemalang, 07 Juli 2001
3. Alamat Rumah : Banyumudal, RT. 02/01,
Moga, Pemalang
4. HP : 085326806988
5. Email : Dhiyaulallam@gmail.com

B. Riwayat Pendidikan

1. Madrasah Ibtidaiyah (MI) Dewi Masyithoh 02 Moga
2. Madrasah Tsanawiyah (MTS) Ihsaniyah Moga
3. Sekolah Menengah Kejuruan (SMK) Al-Falah Moga

Semarang, 20 Juni 2024



Muhammad Dhiyaul Allam

NIM. 1908096047