

Perancangan dan Implementasi Aplikasi Muslim Multiplatform Muslim.in Berbasis Fitur Psychological Chatbot

1st Imaduddin Abdul Karim
AI Center

Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia

imadaka@student.telkomuniversity.ac.id

2nd Suryo Adhi Wibowo
AI Center

Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia

suryoadhiwibowo@telkomuniveristy.ac.id

3rd Thomhert Suprpto Siadari
Fakultas Teknik Elektro

Universitas Telkom
Bandung, Indonesia

thomhert@telkomuniversity.ac.id

Abstrak— Salah satu masalah yang hampir dialami oleh semua orang dan disebabkan karena kesepian dan tidak memiliki tempat untuk curhat adalah kesehatan mental. Sebagian orang sulit memiliki hubungan sosial yang bermakna. Orang-orang tersebut berada dalam resiko kesepian dan menyebabkan masalah kesehatan mental mereka. *Chatbot* sebagai salah satu solusi dalam mengurangi masalah mental berbasis teknologi. *Muslim.in* hadir sebagai aplikasi yang menyediakan fitur *chatbot* yang dapat menjadi tempat bagi mahasiswa untuk menyampaikan keluh kesah mereka (curhat) juga dilengkapi dengan berbagai macam fitur islami untuk menunjang kualitas hidup umat beragama islam. Dengan banyaknya fitur, dibutuhkan sebuah pengujian yang menguji kebergunaan dari aplikasi ini. Metode pengujian yang digunakan adalah *System Usability Scale* (SUS).

SUS adalah alat ukur untuk mengevaluasi kegunaan sistem atau produk berbasis teknologi. Pengujian dengan metode SUS ini menggunakan google form karena lebih efisien dan praktis untuk mengumpulkan data dari responden. Dari 30 responden dapat diperoleh nilai rata-rata yaitu 87,75. Berdasarkan *Percentile Range*, nilai tersebut mendapatkan *grade A*. Sedangkan pada pengujian yang kedua, *Acceptability Ranges* dari nilai 87,75 termasuk *Acceptable*. *Grade Scale* dari nilai 87,75 termasuk ke dalam *grade B*. Sedangkan *Adjective Ratings* dari nilai 87,75 termasuk ke dalam kategori *Excellent*. Dapat disimpulkan bahwa aplikasi *Muslim.in* memiliki nilai yang baik dan layak digunakan.

Kata kunci— Android, Flutter, iOS, Multiplatform, SUS.

I. PENDAHULUAN

Salah satu masalah yang hampir dialami oleh semua orang adalah kesehatan mental. Penyebab masalah kesehatan mental ini salah satunya adalah karena kesepian dan tidak memiliki tempat untuk curhat. Sebagian orang sulit memiliki hubungan sosial yang bermakna. Orang-orang tersebut berada dalam resiko kesepian dan menyebabkan masalah kesehatan mental mereka [1]. Banyak solusi dari teknologi yang telah dikembangkan untuk mengurangi masalah kesehatan mental seseorang. *Chatbot* sebagai salah satu solusi dalam mengurangi masalah mental berbasis teknologi.

Oleh karena masalah-masalah tersebut maka penulis merancang dan mengimplementasikan sebuah aplikasi yang diberi nama *Muslim.in*. *Muslim.in* hadir sebagai aplikasi yang menyediakan fitur *chatbot* yang dapat menjadi tempat

bagi mahasiswa untuk menyampaikan keluh kesah mereka (curhat). Dengan menggunakan teknologi *Artificial Intelligence*, *chatbot* dalam aplikasi *Muslim.in* dapat memberikan respon yang *friendly* layaknya manusia. Selain memiliki fitur *chatbot*, aplikasi *Muslim.in* juga dilengkapi dengan berbagai macam fitur islami untuk menunjang kualitas hidup umat beragama islam. Dengan banyaknya fitur pada aplikasi ini, maka dibutuhkan sebuah pengujian yang menguji kebergunaan dari aplikasi ini. Metode pengujian yang digunakan adalah *System Usability Scale* (SUS).

SUS adalah alat ukur untuk mengevaluasi kegunaan sistem atau produk berbasis teknologi. Keunggulannya yang sederhana dan mudah digunakan, menjadikan SUS menjadi metode yang sering digunakan dalam penelitian dan evaluasi kegunaan produk. Skor SUS memberikan informasi yang berguna untuk menemukan masalah kegunaan dan membandingkan kegunaan sistem yang berbeda.

II. KAJIAN TEORI

Kajian teori yang digunakan pada perancangan aplikasi *Muslim.in* meliputi *Use Case* dan *Activity Diagram*. *Flutter Framework* yang digunakan untuk mengimplementasi *Front-End* dan *Back-End* aplikasi. Serta pengujian aplikasi menggunakan *System Usability Scale* (SUS). Berikut adalah penjelasan dari kajian teori yang digunakan:

A. Use Case Diagram

Use case diagram adalah jenis diagram perilaku dalam *Unified Modeling Language* (UML) yang digunakan untuk menggambarkan interaksi antara pengguna atau sistem eksternal dengan sistem. Biasanya digunakan dalam analisis sistem dan pengembangan perangkat lunak karena memungkinkan pengguna akhir untuk melihat persyaratan fungsional suatu sistem dari sudut pandang pengguna akhir. *Use case diagram* dapat membantu dalam memahami perilaku sistem secara keseluruhan dan berbagai cara pengguna dapat berinteraksi dengannya [2]–[4].

B. Activity Diagram

Diagram aktivitas, juga dikenal sebagai "diagram aktivitas", adalah jenis diagram dalam *Unified Modeling Language* (UML) yang digunakan untuk menunjukkan alur proses atau aliran kerja dari suatu sistem atau proses bisnis. Ini membantu

untuk menunjukkan urutan aktivitas, keputusan, dan paralelisme yang ada dalam sebuah proses. Secara keseluruhan, diagram aktivitas adalah cara yang bagus untuk menyajikan proses atau alur kerja dengan cara yang sederhana dan mudah dipahami sehingga dapat menganalisis, mendesain, dan berkomunikasi tentang bagaimana suatu sistem atau proses bekerja [5].

C. Flutter Framework

Flutter adalah kerangka kerja (*frame work*) *open source* yang dikembangkan oleh Google yang memungkinkan pengembang membuat aplikasi yang cantik, cepat, dan responsif dengan menggunakan *source code* yang sama untuk berbagai platform. Dalam bahasa yang lebih sederhana, Flutter adalah alat yang memungkinkan pengembang membuat aplikasi lintas platform, seperti aplikasi mobile (Android dan iOS), web, dan desktop. *Framework* ini menggunakan bahasa pemrograman Dart [6].

D. Front-End

Front-end adalah bagian dari sebuah aplikasi atau situs web yang terlihat dan dirasakan oleh pengguna saat mereka menggunakan perangkat mereka. Ini berfokus pada antarmuka pengguna (UI) dan bagaimana informasi disajikan dan diatur sehingga mudah dipahami dan digunakan. Selain itu, *front-end* harus memastikan bahwa aplikasi atau situs web dapat diakses dan beroperasi dengan baik di berbagai platform, seperti komputer desktop, tablet, dan ponsel pintar. Ini memastikan bahwa pengguna memiliki pengalaman yang konsisten dan responsif di berbagai platform [6].

E. Back-End

Bagian di belakang layar sebuah aplikasi atau situs web yang tidak terlihat oleh pengguna secara langsung disebut *back-end*. Bagian ini bertanggung jawab untuk mengelola dan mengatur data serta logika bisnis aplikasi, serta bagaimana data disimpan, dikelola, dan diproses. Bagian ini juga bertanggung jawab atas cara aplikasi berinteraksi dengan *database* dan sumber daya lainnya. *Back-end* juga mengelola performa aplikasi, yang memastikan situs web atau aplikasi berjalan dengan cepat. Pengembang *back-end* harus memastikan bahwa infrastruktur dan *server* yang digunakan mampu menangani beban pengguna yang luar biasa dan beradaptasi dengan baik [6].

F. Human Computer Interaction (HCI)

HCI adalah cabang ilmu komputer tentang interaksi antara manusia dan sistem komputer. HCI mencakup studi, desain, pengembangan, dan evaluasi antarmuka pengguna komputer, dengan mempertimbangkan aspek psikologis, sosiologis, dan teknis. Tujuannya untuk membuat user interface yang intuitif, efisien, dan fungsional antara pengguna dan komputer. Ini membutuhkan pemahaman tentang kebutuhan, perilaku, dan preferensi pengguna saat menggunakan sistem komputer. Dalam bidang ini, pengguna dianggap sebagai faktor penting dalam perancangan dan pengembangan teknologi [7].

G. System Usability Scale (SUS)

SUS adalah alat ukur untuk mengevaluasi kegunaan sistem atau produk berbasis teknologi [8]. SUS banyak digunakan dalam *Human Computer Interaction* (HCI) dan desain interaksi untuk mengukur bagaimana pengguna me-

mandang suatu sistem agar mudah digunakan [7]. SUS memiliki keunggulan karena sederhana dan mudah digunakan, sehingga sering digunakan untuk penelitian dan evaluasi kegunaan suatu produk. Skor SUS memberikan informasi yang berguna untuk mengidentifikasi masalah kegunaan dan membandingkan kegunaan dari sistem yang berbeda.

Pengujian dengan metode SUS ini menggunakan google form karena lebih efisien dan praktis untuk mengumpulkan data dari responden. Langkah pertama yang dilakukan adalah dengan membuat formulir di google form yang berisi 10 pernyataan yang akan dinilai oleh responden dengan skala Likert. Terdapat 5 pernyataan yang bernilai positif dan 5 bernilai negatif. Pernyataan-pernyataan bernomor ganjil (1,3,5,7, dan 9) merupakan pernyataan yang bernilai positif, sedangkan pernyataan-pernyataan bernomor genap (2,4,6,8, dan 10) bernilai negatif. 10 pernyataan tersebut antara lain [9]:

1. Saya berpikir akan sering menggunakan sistem ini lagi.
2. Saya merasa sistem ini rumit untuk digunakan.
3. Saya merasa sistem ini mudah digunakan.
4. Saya membutuhkan bantuan dari orang lain atau teknisi dalam menggunakan sistem ini.
5. Saya merasa fitur-fitur pada sistem ini berjalan dengan semestinya.
6. Saya merasa ada banyak hal yang tidak konsisten (tidak serasi pada sistem ini).
7. Saya merasa orang lain akan memahami cara menggunakan sistem ini dengan cepat.
8. Saya merasa sistem ini membingungkan.
9. Saya merasa tidak ada hambatan saat menggunakan sistem ini.
10. Saya perlu membiasakan diri terlebih dahulu sebelum menggunakan sistem ini.

Setelah itu, formulir tersebut dikirimkan kepada 30 responden untuk mengevaluasi aplikasi Muslim.in. Responden diminta untuk menilai setiap pernyataan dalam formulir menggunakan skala *Likert*. Skala *Likert* ini memiliki rentang nilai 1-5, dengan nilai 1 berarti sangat tidak setuju, 2 berarti tidak setuju, 3 berarti ragu-ragu, 4 berarti setuju, dan 5 berarti sangat setuju [9].

Total nilai yang diberikan oleh masing-masing responden akan dihitung terlebih dahulu menggunakan rumus (1) [6]:

$$x = ((P1 - 1) + (5 - P2) + (P3 - 1) + (5 - P4) + (P5 - 1) + (5 - P6) + (P7 - 1) + (5 - P8) + (P9 - 1) + (5 - P10)) * 2,5 \quad (1)$$

Keterangan:

x = total nilai tiap responden.

P = nilai dari setiap pernyataan.

Setelah semua nilai dari tiap responden didapatkan. Selanjutnya adalah dengan menghitung nilai rata-rata dari seluruh responden dengan menggunakan rumus (2) [9]:

$$\bar{x} = \frac{\sum x}{n} \quad (2)$$

Keterangan:

\bar{x} = rata-rata total nilai.

n = jumlah responden.

III. PERANCANGAN SISTEM

Perancangan sistem yang dilakukan pada perancangan aplikasi Muslim.in meliputi perancangan *Use Case* dan *Activity Diagram*. Implementasi *Front-End* dan *Back-End* dari aplikasi Muslim.in. Berikut adalah penjelasan dari perancangan *Use Case*, *Activity Diagram*, *Front-End*, dan *Back-End*:

A. Use Case Diagram

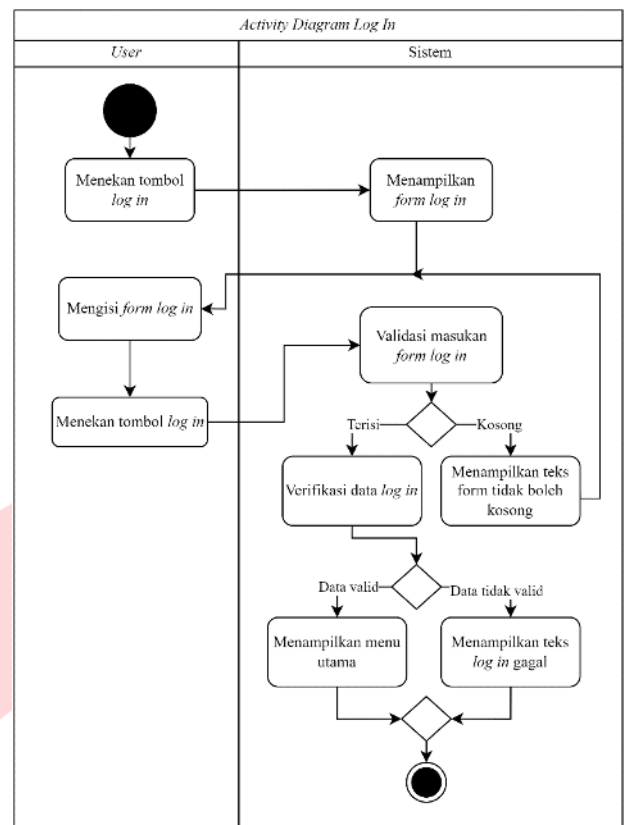
Pada Gambar 1 terlihat bahwa dua faktor yaitu *user* dan *admin*. Sebelum menggunakan aplikasi, *user* diminta untuk registrasi dan *login*. Namun, *user* juga bisa masuk ke dalam aplikasi tanpa memiliki akun (*log in as guest*). *User* dapat mengakses berbagai macam fitur yang terdapat pada aplikasi seperti *chatbot*, jadwal salat, musholla, resto halal, al-ma'tsurat (pagi dan petang), al-qur'an, tasbih, dan qiblat. Sementara itu, *admin* diperlukan untuk melakukan *maintenance* aplikasi serta melakukan pembaruan pada data dan model *deep learning chatbot*.



GAMBAR 1
Use Case Diagram

B. Activity Diagram

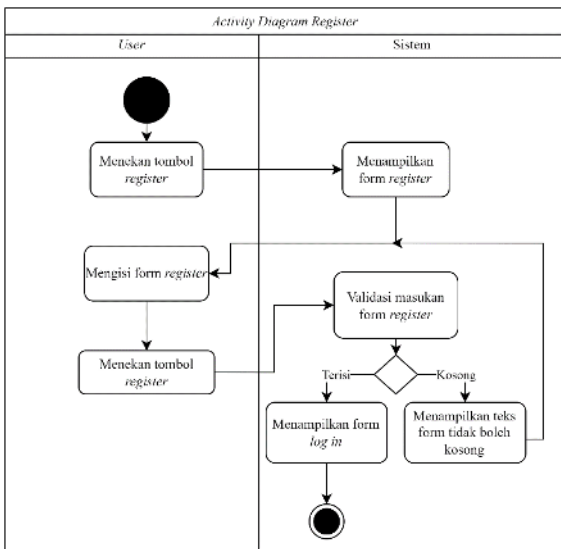
Pada Gambar 2, menjelaskan tentang *activity diagram log in* yang dimulai dengan *user* menekan tombol *log in*. Setelah tombol ditekan, sistem akan menampilkan *form log in* yang berisi *email* dan *password* untuk kemudian diisi oleh *user*. Setelah mengisi dan menekan tombol *log in*, Sistem akan melakukan validasi terhadap data yang masukkan oleh *user*. Apabila datanya kosong, maka akan ditampilkan teks bahwa *form* tidak boleh kosong. Jika data terisi, data akan dicek apakah antara *email* dan *password* sesuai. Apabila tidak sesuai, sistem akan menampilkan teks *log in* gagal.



GAMBAR 2
Activity Diagram 1

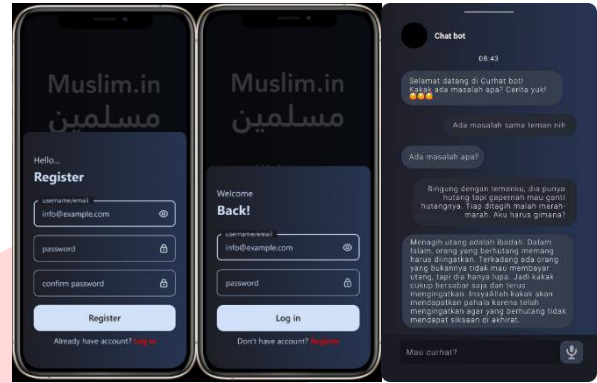
Pada *activity diagram register* yang ditunjukkan oleh Gambar 3, *user* dapat melakukan registrasi akun dengan menekan tombol *register*. Setelah tombol ditekan, sistem akan menampilkan *form* registrasi akun yang kemudian akan diisi oleh *user*. Setelah data diisi, data yang dimasukkan akan divalidasi. Jika datanya kosong, sistem akan menampilkan teks bahwa *form* tidak boleh kosong. Jika datanya terisi, sistem akan menampilkan *form log in* untuk masuk ke dalam aplikasi.

Pada Gambar 4 di bawah, ditunjukkan *activity diagram* untuk fitur *chatbot*. Pengguna dapat menggunakan *chatbot* dengan menekan *bar chatbot*. Kemudian, aplikasi akan menampilkan halaman *chatting*. Pengguna dapat merekam suara ataupun menuliskan pesannya pada tempat yang tersedia. Setelah pesan dimasukkan oleh pengguna, aplikasi akan mengirimkan teks yang dimasukkan ke filter *chatbot* untuk dianalisis apakah teks tersebut termasuk curhat atau bukan curhat. Jika teks termasuk ke dalam curhat, maka teks akan dikirimkan ke API OpenAI untuk mendapatkan pesan balasan dari *chatbot* yang kemudian akan ditampilkan kembali ke pengguna. Jika teks yang dimasukkan tidak termasuk curhat, maka aplikasi akan menampilkan pesan bahwa teks yang dimasukkan bukan curhat.

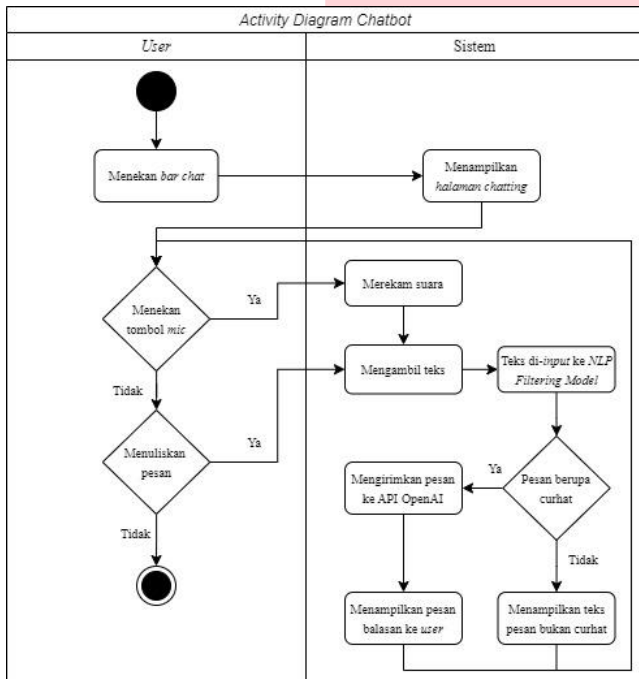


GAMBAR 3 Activity Diagram 2

widget dalam tata letak yang diinginkan. Tampilan yang dibangun juga responsif dan dapat disesuaikan dengan berbagai ukuran layar dan perangkat. Hal ini akan menjamin pengalaman menggunakan aplikasi yang konsisten dan nyaman pada berbagai platform. Setelah pembangunan front-end selesai, pengujian akan dilakukan untuk memastikan bahwa tampilan dan kemampuan aplikasi memenuhi harapan.



GAMBAR 5 Perancangan Ui/Ux



GAMBAR 4 Activity Diagram 3

C. Perancangan Aplikasi (Front-End dan Back-End)

Front-end, yang dikembangkan menggunakan bahasa pemrograman Dart, adalah bagian dari aplikasi Flutter yang berhubungan langsung dengan pengguna dan bertanggung jawab untuk menampilkan antarmuka pengguna (UI) yang menarik dan responsif. Langkah pertama adalah membuat antarmuka pengguna (UI) dan pengalaman pengguna (UX) aplikasi dengan menggunakan aplikasi figma. Gambar 5 menunjukkan hasil desain UI/UX. Setelah desain UI/UX selesai, langkah berikutnya adalah memilih widget yang tepat untuk membuat tampilan yang sesuai dengan desain. Widget bawaan Flutter termasuk Container, Text, Image, dan Button. Widget-widget tersebut digunakan sesuai dengan desain yang telah dirancang. Dalam implementasi ini, bahasa pemrograman Dart digunakan untuk menulis kode dan mengatur

Back-end aplikasi Flutter adalah komponen yang tidak terlihat oleh pengguna secara langsung, tetapi berfungsi sebagai server yang mengelola data, menyimpan data, dan mengatur alur kerja aplikasi. Back-end pada aplikasi ini, menggunakan service Firebase, HTTP Request ke Virtual Machine, dan API-API dari Framework Flutter. Langkah pertama saat menggunakan service Firebase adalah mendapatkan file konfigurasi Firebase, seperti google-services.json untuk Android atau GoogleService-Info.plist untuk iOS. File-file ini mengandung informasi yang diperlukan oleh aplikasi Flutter untuk terhubung dengan proyek Firebase yang telah dibuat. Setelah konfigurasi Firebase berhasil, selanjutnya adalah menghubungkan layanan Firebase Auth untuk akun pengguna, Firestore untuk penyimpanan data pengguna dan resto halal, dan Fire Storage untuk menampilkan gambar resto halal. Model deep learning pada fitur chatbot disimpan dan dijalankan di Virtual Machine. Untuk menggunakannya dibutuhkan HTTP request agar pengguna dapat menggunakan model deep learning tersebut. Metode yang digunakan saat HTTP request adalah POST. Metode POST digunakan untuk mengirimkan data curhatan ke server. Untuk fitur-fitur islami seperti, Arah Qiblat, Jadwal Sholat, Al-Qur'an, dan sebagainya menggunakan API-API yang telah disediakan oleh komunitas Flutter.

IV. HASIL DAN PEMBAHASAN

Keluaran dari perancangan aplikasi ini adalah aplikasi multiplatform yang bisa dijalankan di platform iOS dan Android yang diberi nama Muslim.in. Aplikasi ini diuji menggunakan metode System Usability Scale (SUS) untuk menguji kegunaan dan kelayakan dari aplikasi Muslim.in. Berikut merupakan hasil implementasi dan pembahasan mengenai pengujian aplikasi Muslim.in:

A. Hasil Implementasi

Hasil akhir pada penelitian ini adalah aplikasi muslim multiplatform yang memiliki fitur chatbot curhat dan dapat diunduh melalui Google Play Store dan App Store. Tampilan dari aplikasi Muslim.in pada platform Android dapat dilihat pada Gambar 6. Sedangkan pada platform iOS dapat dilihat pada Gambar 7. Implementasi yang dilakukan adalah berupa implementasi antarmuka. Implementasi antarmuka dilakukan pada setiap fitur maupun program yang telah dibangun dan diimplementasikan dengan pengkodean dalam bentuk file program dengan Bahasa dart. Tabel 1 di bawah adalah implementasi antarmuka dari fitur-fitur tambahan. Pada fitur Chatbot akan ditampilkan pada halaman utama “HomePage”, dan file program disimpan dengan nama file “view/chatbot.dart”. Pada tampilan fitur chatbot ini akan ada pesan sambutan dari bot. Pada bagian bawah aplikasi terdapat textfield yang berfungsi untuk mengetikkan curhatan pengguna. Pada bagian samping textfield terdapat tombol mikrofon yang akan otomatis berubah menjadi tombol kirim jika pengguna telah mengetikkan curhatan di textfield. User akan menuliskan atau merekam suara untuk mengirimkan curhat. Teks tersebut akan dimasukkan ke dalam Virtual Machine (VM) menggunakan HTTP request. VM akan mengirimkan respons dari data curhat yang dikirimkan pengguna. Semua data yang dikirimkan pada alur ini berupa JSON.



GAMBAR 6 Hasil Aplikasi Di Android



GAMBAR 7 Hasil Aplikasi Di Ios

TABEL 1 Hasil Implementasi

No	Fitur	Deskripsi	Nama File
1	Halaman Al-Qur'an	Halaman ini menampilkan nama-nama surah berdasarkan urutan surah dalam Al-Qur'an	view/Quran/quran.dart

No	Fitur	Deskripsi	Nama File
2	Halaman Al-Qur'an - Surah	Halaman ini menampilkan ayat-ayat Al-Qur'an dari surah yang dipilih.	view/Quran/quran_content.dart
3	Halaman setting Al-Qur'an	Halaman ini berisi pengaturan tampilan Al-Qur'an.	view/Quran/settings.dart
4	Halaman Al-Ma'tsurat	Halaman ini akan menampilkan bacaan dzikir yang telah dipilih user.	view/almatsurat.dart
5	Halaman Resto Halal	Halaman ini menampilkan restoran-restoran halal yang ada di Korea, terdapat fitur pencarian dan filter.	view/Resto/Resto.dart
6	Fitur tanggal Masehi, Hijriyah, dan Jadwal salat	Ketiga fitur ini ditampilkan di halaman utama (Home Page)	view/homepage.dart dan view/guesthome.dart
7	Fitur Musholla	Fitur ini akan otomatis membuka aplikasi google maps dan memberikan rekomendasi Masjid terdekat.	view/homepage.dart dan view/guesthome.dart
8	Halaman Tasbih	Halaman ini menampilkan jumlah dzikir yang dilakukan user, yang jumlahnya dihitung dari setiap sentuhan yang dilakukan oleh user di area yang telah disediakan.	view/tasbih.dart
9	Halaman Qiblat	Halaman ini akan menampilkan arah qiblat di lokasi user dan akan menampilkan posisi derajat dari utara.	view/qiblat_screen.dart
10	Landing Page	Halaman yang memuat deskripsi singkat aplikasi dan halaman log in atau registrasi sesuai yang dipilih user.	view/Account/landingpage.dart
11	Log in Widget	Berada di halaman Landing Page, halaman ini akan meminta pengguna memasukkan e-mail dan password yang telah terdaftar.	view/Account/loginwidget.dart
12	Halaman Registrasi	Berada di halaman Landing Page, halaman ini akan meminta pengguna memasukkan e-mail username, dan password.	view/Account/signinwidget.dart
13	Home Page	Halaman utama yang menampilkan username, fitur-fitur, dan tombol-tombol untuk mengakses beberapa fitur.	view/homepage.dart
14	Side Menu	Halaman yang berisi informasi user seperti username dan e-mail, serta pilihan untuk log out.	view/sidemenu.dart
15	Guest Home Page	Halaman utama bagi pengguna yang tidak mempunyai akun yang fitur-fitur, dan tombol-tombol untuk mengakses beberapa fitur.	view/guesthome.dart
16	Splash Screen	Halaman awal yang memuat logo aplikasi	view/splash.dart
17	Forgot Password	Halaman yang berguna untuk memulihkan akun pengguna jika lupa password.	view/Account/forgotpassword.dart

B. Pengujian System Usability Scale (SUS)

Dengan menggunakan data yang didapatkan dari formulir yang telah diisi oleh 30 responden, data tersebut diolah menggunakan kedua rumus yang telah di bahas di bab III

Metode. Hasil rata-rata total nilai dari seluruh responden adalah 87,75 yang dapat dilihat pada Tabel 2.

TABEL 2
Hasil Responden

No	Responden	Total Nilai (x)
1	Faras Syahrani Dalimunthe	100
2	Siti Nurkhasanah	100
3	Luthfi Kurniawan	97,5
4	Rusmayanti	95
5	Sopandi	100
6	Ayu Andini	100
7	Aldira Fadillah Lazuardi	95
8	Ferdian Ilham Ramadhan	100
9	Reynaldi	72,5
10	Arik	97,5
11	Mukhlis Kurnia Bakti	87,5
12	Fabiano Abbey Karo Sekali	95
13	Rizqi Syah	65
14	Dhiya Fathiyya	80
15	Fargy Septiari Munir	97,5
16	Alvaro Septra Domingo Nauw	90
17	Prita Trivena Hesmono	77,5
18	MARYAM BURHANUDDIN	100
19	Iga	90
20	Mutiara Balqis	95
21	Fadilah Muhshi Al Qodar	70
22	M Yayan Abdullah	62,5
23	Farras erdiansyah	80
24	Yuwan	100
25	Andi Wahyu Maulana	100
26	Arif Faturrachman	85
27	Rakha Yusan	77,5
28	Dina Khairunisa	87,5
29	Jihan Nur Alifia Hrp	80
30	Seintya gimi lestari	55
Rata-rata Total Nilai		87,75

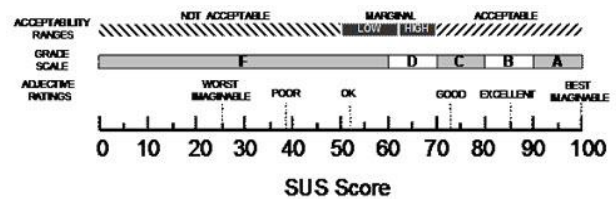
Baik tidaknya nilai dari suatu SUS dapat dilihat dari rata-rata total nilai yang didapatkan. Nilai rata-rata tersebut didapatkan dengan cara menjumlahkan total nilai dari seluruh responden lalu dibagi dengan jumlah responden. Pada pengujian kali ini terdapat 30 orang responden. Dari 30 orang responden tersebut didapatkan nilai rata-rata total nilai sebesar 87,75. Untuk menguji baik tidaknya nilai tersebut maka terdapat 2 cara yang dapat digunakan. Cara yang pertama adalah dengan melihat SUS percentile range score. Percentile range ini memiliki 5 grade penilaian yang terdiri dari A, B, C, D, dan E. Perbedaan dari tiap grade dapat dilihat pada Tabel 3 [9].

TABEL 3
ANALISIS BERDASARKAN PERCENTILE RANGE

Grade	Range Rata-rata Total Nilai (\bar{x})
A	$\bar{x} \geq 80,3$
B	$74 \leq \bar{x} < 80,3$
C	$68 \leq \bar{x} < 74$
D	$51 \leq \bar{x} < 68$
E	$\bar{x} < 51$

Cara kedua adalah dengan menentukan Acceptability Ranges, Grade Scale, dan Adjective Ratings. Acceptability Ranges memiliki 3 (tiga) kategori yaitu Not Acceptable, Marginal, dan Acceptable. Grade Scale memiliki 5 kategori yaitu A, B, C, D, dan F. Sedangkan Adjective Ratings

memiliki 6 kategori yaitu Worst Imaginable, Poor, Ok, Good, Excellent, dan Best Imaginable. Ketiga hal tersebut akan diukur menggunakan rata-rata total nilai SUS yang sudah didapatkan yaitu 87,75. Untuk lebih jelasnya dapat dilihat pada Gambar 8 di bawah [10].



GAMBAR 8
Perbandingan Acceptability Ranges, Grade Scale, Dan Adjective Ratings Terhadap Sus Score

Dari 30 responden dapat diperoleh nilai rata-rata yaitu 87,75. Pada cara pengujian yang pertama yaitu menentukan Grade dari Percentile Range, jika dilihat pada Tabel 2 nilai 87,75 berada pada grade A karena nilai tersebut lebih besar dari 80,3. Sedangkan pada pengujian yang kedua, Acceptability Ranges dari nilai 87,75 termasuk Acceptable karena nilai tersebut lebih besar dari 70. Grade Scale dari nilai 87,75 termasuk ke dalam grade B karena berada pada rentang nilai 80-90. Sedangkan Adjective Ratings dari nilai 87,75 termasuk ke dalam kategori Excellent karena memiliki nilai lebih dari 85. Dari seluruh hasil tersebut dapat disimpulkan bahwa aplikasi Muslim.in memiliki nilai yang baik dan layak untuk digunakan.

V. KESIMPULAN

Dari perancangan dan implementasi yang didapatkan, dapat disimpulkan bahwa aplikasi Muslim.in telah berhasil dirancang dan dapat terintegrasi dengan Virtual Machine, Cloud Storage, dan API-API dengan baik. SUS Score dari total 30 responden yang didapat sebesar 87,75. Pada cara pengujian yang pertama yaitu menentukan Grade dari Percentile Range, nilai 87,75 berada pada grade A karena nilai tersebut lebih besar dari 80,3. Sedangkan pada pengujian yang kedua, Acceptability Ranges dari nilai 87,75 termasuk Acceptable karena nilai tersebut lebih besar dari 70. Grade Scale dari nilai 87,75 termasuk ke dalam grade B karena berada pada rentang nilai 80-90. Sedangkan Adjective Ratings dari nilai 87,75 termasuk ke dalam kategori Excellent karena memiliki nilai lebih dari 85. Dari seluruh hasil tersebut dapat disimpulkan bahwa aplikasi Muslim.in memiliki nilai yang baik dan layak untuk digunakan.

REFERENSI

- [1] C. M. Masi, H. Y. Chen, L. C. Hawkey, dan J. T. Cacioppo, "A meta-analysis of interventions to reduce loneliness," *Personality and Social Psychology Review*, vol. 15, no. 3. SAGE Publications Inc., hlm. 219–266, 2011. doi: 10.1177/1088868310377394.
- [2] L. Setiyani, "Desain Sistem: Use Case Diagram," dalam *Prosiding Seminar Nasional Inovasi dan Adopsi Teknologi (INOTEK)*, 2021, hlm. 246–260.
- [3] A. Y. Aleryani, "Comparative study between data flow diagram and use case diagram," *International Journal of Scientific and Research Publications*, vol. 6, no. 3, hlm. 124–126, 2016.

- [4] S. S. Mule, Y. Waykar, dan S. V. Mahavidyalaya, "Role of USE CASE diagram in s/w development," *International Journal of Management and Economics*, 2015.
- [5] J. W. Satzinger, R. B. Jackson, dan S. D. Burd, *Systems analysis and design in a changing world*. Cengage learning, 2015.
- [6] B. Raharjo, "Pemrograman Android dengan Flutter," *Bandung: Informatika Bandung*, 2019.
- [7] A. Bangor, P. T. Kortum, dan J. T. Miller, "An Empirical Evaluation of the System Usability Scale," *Int J Hum Comput Interact*, vol. 24, no. 6, hlm. 574–594, Jul 2008, doi: 10.1080/10447310802205776.
- [8] J. Brooke, "Sus: a "quick and dirty" usability," *Usability evaluation in industry*, vol. 189, no. 3, hlm. 189–194, 1996.
- [9] A. Saputra, "Penerapan Usability pada Aplikasi PENTAS Dengan Menggunakan Metode System Usability Scale (SUS)," *JTIM : Jurnal Teknologi Informasi dan Multimedia*, vol. 1, no. 3, hlm. 206–212, Nov 2019, doi: 10.35746/jtim.v1i3.50.
- [10] A. Bangor, P. Kortum, dan J. Miller, "Determining what individual SUS scores mean: Adding an adjective rating scale," *J Usability Stud*, vol. 4, no. 3, hlm. 114–123, 2009.