

**Implementasi Personalisasi dengan Menggunakan Sesi
untuk Menyimpan Data Hafalan Pengguna dalam
Aplikasi Al-Qur'an Berbasis Mobile**



Disusun Oleh:

Nama : Ananda Shafa Risha Waluyo

NIM : 19523231

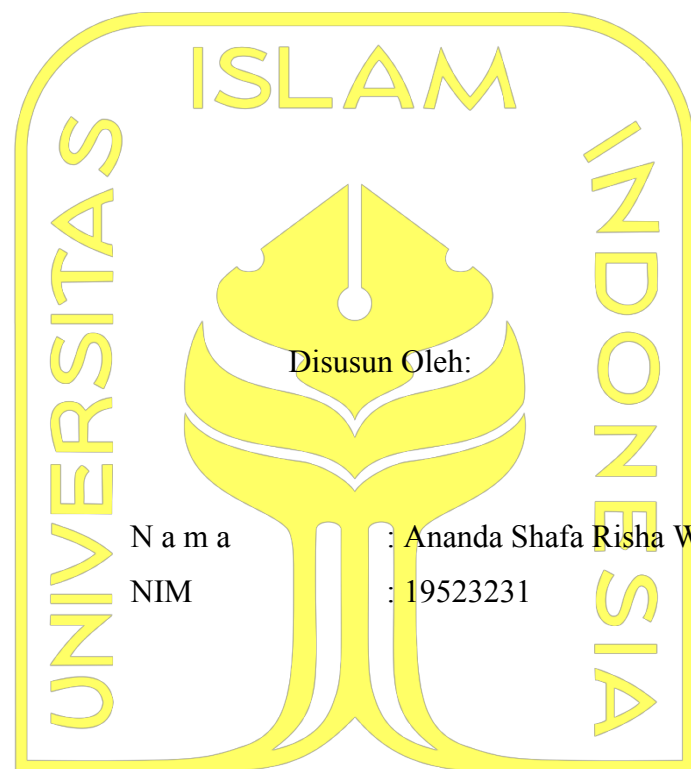
**PROGRAM STUDI INFORMATIKA – PROGRAM SARJANA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA**

2024

HALAMAN PENGESAHAN DOSEN PEMBIMBING

**Implementasi Personalisasi dengan Menggunakan Sesi
untuk Menyimpan Data Hafalan Pengguna dalam
Aplikasi Al-Qur'an Berbasis Mobile**

TUGAS AKHIR



الجمعة الإسلامية الأندونيسية

Yogyakarta, 6 Januari 2024

Pembimbing,

(Kholid Haryono, S.T., M.Kom)

HALAMAN PENGESAHAN DOSEN PENGUJI

**Implementasi Personalisasi dengan Menggunakan Sesi
untuk Menyimpan Data Hafalan Pengguna dalam
Aplikasi Al-Qur'an Berbasis Mobile**

TUGAS AKHIR

Telah dipertahankan di depan sidang penguji sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer dari Program Studi Informatika – Program Sarjana di Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 6 Januari 2024

Tim Penguji

Kholid Haryono, S.T., M.Kom.

Anggota 1

Elyza Gustri Wahyuni, S.T., M.Cs.

Anggota 2

Sri Mulyati, S.Kom., M.Kom.

Mengetahui,

Ketua Program Studi Informatika – Program Sarjana

Fakultas Teknologi Industri

Universitas Islam Indonesia

(Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D.)

HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : Ananda Shafa Risha Waluyo
NIM : 19523231

Tugas akhir dengan judul:

Implementasi Personalisasi dengan Menggunakan Sesi untuk Menyimpan Data Hafalan Pengguna dalam Aplikasi Al-Qur'an Berbasis Mobile

Menyatakan bahwa seluruh komponen dan isi dalam tugas akhir ini adalah hasil karya saya sendiri. Apabila di kemudian hari terbukti ada beberapa bagian dari karya ini adalah bukan hasil karya sendiri, tugas akhir yang diajukan sebagai hasil karya sendiri ini siap ditarik kembali dan siap menanggung risiko dan konsekuensi apapun.

Demikian surat pernyataan ini dibuat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 11 Januari 2024



(Ananda Shafa Risha Waluyo)

HALAMAN PERSEMBAHAN

Alhamdulillah, ya Allah SWT, puji syukur ke hadiratmu tuhan semesta alam penulis sampaikan. Karena dengan kenikmatan yang engkau berikan penulis dapat melewati segala tantangan dan rintangan dalam menghadapi kehidupan di dunia ini. Dengan rahmatmu juga penulis dapat mampu untuk menyelesaikan studi strata satu (S1) Jurusan Informatika Universitas Islam Indonesia.

Penulis mengucapkan banyak terima kasih untuk kedua orang tua penulis yang sudah bekerja keras untuk penulis, tanpa ada rasa pamrih. Kedua orang tua penulis yang selalu mendukung dan mendorong penulis untuk dapat selalu bekerja keras dan berkembang sampai penulis dapat berada pada titik ini. Rasa sayang kedua orang tua yang sangat melimpah tidak akan pernah penulis mampu untuk bayar walaupun habis waktu penulis, untuk itu penulis sangat berterima kasih.

Untuk dosen-dosen ku yang selalu ada dalam menemani, studi penulis, penulis ucapkan terima kasih banyak. Dosen-dosen yang selalu sabar dan membuat penulis berkembang, namun penulis tidak dapat membalas jasa mereka. Dan juga seluruh civitas kampus Universitas Islam Indonesia.

Untuk teman-teman tetaplah bekerja keras walaupun akan ada saatnya kita berada di bawah janganlah sekali-kali menyerah. Kesempatan akan selalu hadir untuk setiap orang yang berusaha. Ingatlah usaha tidak akan mengkhianati hasil, jadi berusahalah sebaik mungkin walaupun menerima cibiran, anggaplah itu sebagai motivasi.

HALAMAN MOTO

“It Shall be done”

-King Arthas

KATA PENGANTAR

Assalamu'alaikum wr. wb.

Alhamdulillah, dengan rahmat Allah yang senantiasa dilimpahkan kepada umatnya, penulis dapat mampu untuk menyelesaikan tugas akhir ini. Meskipun ada halangan dan rintangan yang tidak diduga. Namun, atas berkah dan rahmat Allah penulis akhirnya dapat melewati halangan dan rintangan tersebut.

Penulis ingin mengucapkan rasa terima kasih penulis kepada pihak-pihak yang selama ini terus mendukung penulis hingga akhirnya, penulis dapat menyelesaikan tugas akhir ini. Penulis mengucapkan rasa terima kasih sebesar mungkin kepada:

1. Orang tua saya yang selalu ada, dan memberikan semangat, walaupun saya tidak meminta apapun, mereka selalu ada.
2. Bapak dosen pembimbing saya bapak Kholid Haryono yang selalu sabar dan mau membantu saya.
3. Kepada teman-teman penulis yang selalu menemani penulis dalam suka dan duka, dan selalu ada dalam mengatasi crisi dan kesulitan bersama.

Semoga rahmat Allah selalu dilimpahkan untuk kita semua. Akhir kata semoga tugas akhir penulis dapat memacu inspirasi, dan berkontribusi untuk pengembangan dan penelitian di masa depan.

Yogyakarta, 6 Januari 2024

(Ananda Shafa Risha Waluyo)

SARI

Al-Qur'an diajarkan dan dihafalkan oleh jutaan umat islam di dunia seperti pada awal diturunkan. Namun untuk menghafalkannya bukanlah hal yang mudah dan sederhana. Dengan adanya kesibukan yang terus menerus terjadi dan tingkat memori menjadi sebuah tantangan dalam menghafalkan kitab suci Al-Qur'an. Sehingga, menjadi peluang untuk menyediakan sebuah aplikasi digital pada gawai mereka yang memiliki fungsi untuk memudahkan penghafal untuk dapat melihat catatan *progress* hafalan mereka. Metode yang digunakan adalah *Software Development Life Cycle* (SDLC). Berdasarkan hasil dari pengujian fungsionalitas dan *User Acceptance Test* (UAT) didapatkan Tingkat keberhasilan masing-masing 92,8% dan 88,3%. Dari hasil tersebut dapat dibuktikan bahwa aplikasi sudah memenuhi kebutuhan pengguna.

Kata kunci: Al-Qur'an, *Software Development Life Cycle* (SDLC), *User Acceptance Test* (UAT), hafalan, catatan.

GLOSARIUM

<i>Platform</i>	Merupakan deskripsi general yang merujuk ke sebuah hal yang didukung, dengan contoh <i>platform mobile</i> yang merujuk kepada perangkat yang didukung seperti Android maupun IOS.
<i>Stakeholder</i>	Merupakan pemangku kepentingan dalam sebuah organisasi ataupun perusahaan.
<i>Mobile</i>	Mudah bergerak (berpindah) atau digerakkan (dipindah-pindahkan).
<i>Autentikasi</i>	Proses, cara, perbuatan membuktikan sesuatu secara autentik.
<i>Mushaf</i>	Bagian naskah Al-Qur'an.
<i>Framework</i>	Kerangka kerja yang digunakan untuk mengembangkan <i>software</i>

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGESAHAN DOSEN PEMBIMBING	ii
HALAMAN PENGESAHAN DOSEN PENGUJI	iii
HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR	iv
HALAMAN PERSEMBAHAN	v
HALAMAN MOTO	vi
KATA PENGANTAR	vii
SARI	viii
GLOSARIUM	ix
DAFTAR ISI	x
DAFTAR TABEL	xii
DAFTAR GAMBAR	xiii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Tujuan Penelitian	2
1.5 Manfaat Penelitian	2
1.6 Metodologi Penelitian	2
1.7 Sistematika Penulisan	3
BAB II TINJAUAN PUSTAKA	5
2.1 Penelitian Terdahulu	5
2.2 Kajian Teori	9
2.2.1 Quran Digital	9
2.2.2 Personalisasi	9
2.2.3 Sesi	10
2.3 Pengembangan Aplikasi	10
2.4 Software Development Life Cycle(SDLC)	11
2.4.1 Perencanaan	11
2.4.2 Desain	11
2.4.3 Implementasi	11
2.4.4 Pengujian	11
2.5 Pengembangan Aplikasi mobile	12
2.5.1 Alat-alat pengembangan aplikasi mobile	12
2.5.2 React Native	13
2.5.3 Firebase	13
2.5.4 <i>Unified Modelling Language</i> (UML)	13
2.5.5 Pengujian Sistem	14
2.5.6 <i>Blackbox testing</i>	14
2.5.7 <i>User Acceptance Testing</i> (UAT)	14
BAB III METODOLOGI	16
3.1 Perencanaan	16
3.1.1 Analisis Kebutuhan	16
3.1.2 Perencanaan Alur Proses Sistem	17
3.2 Desain	18
3.2.1 Desain Usecase Diagram	18
3.2.2 Desain Login	19

3.2.3	Desain Register.....	22
3.2.4	Desain Logout	24
3.2.5	Desain Database	26
3.2.6	Desain Download	28
3.3	Implementasi.....	30
3.3.1	Register.....	32
3.3.2	Login	34
3.3.3	Logout	36
3.3.4	Download	37
3.4	Pengujian.....	40
3.4.1	<i>Blackbox Testing</i>	40
3.4.2	<i>User Acceptance Test(UAT)</i>	41
	BAB IV HASIL DAN PEMBAHASAN	42
4.1	Fitur.....	42
4.1.1	Fitur Login.....	42
4.1.2	Fitur Register.....	44
4.1.3	Fitur Logout.....	46
4.1.4	Fitur Download.....	47
4.2	Pengujian.....	47
4.2.1	<i>Blackbox Testing</i>	47
4.2.2	<i>User Acceptance Test (UAT)</i>	49
	BAB V KESIMPULAN DAN SARAN.....	52
5.1	Kesimpulan	52
5.2	Saran.....	52
	DAFTAR PUSTAKA	55
	LAMPIRAN	59

DAFTAR TABEL

Tabel 3. 1 Tabel kebutuhan.....	16
Tabel 4. 1 Tabel hasil uji fungsionalitas.....	48
Tabel 4. 2 Tabel pertanyaan kuesioner	49
Tabel 4. 3 Tabel bobot penilaian kuesioner	50
Tabel 4. 4 Tabel hasil penilaian kuesioner.....	50

DAFTAR GAMBAR

Gambar 3. 1 Gambar alur proses sistem	17
Gambar 3. 2 Gambar usecase pengguna	19
Gambar 3. 3 Gambar activity diagram login	20
Gambar 3. 4 Gambar activity diagram lupa password.....	21
Gambar 3. 5 Gambar protorype figma halaman login	22
Gambar 3. 6 Gambar activity daigram register.....	23
Gambar 3. 7 Gambar prototype figma halaman register.....	24
Gambar 3. 8 Gambar activity diagram logout	25
Gambar 3. 9 Gambar prototype figma halaman pengaturan.....	26
Gambar 3. 10 Gambar arsitektur firebase Sumber:	27
Gambar 3. 11 Gambar relasi database	28
Gambar 3. 12 Gambar activity diagram download.....	29
Gambar 3. 13 Gambar prototype pop-up download	30
Gambar 3. 14 Gambar dependensi penyimpanan	31
Gambar 3. 15 Gambar kode untuk penyimpanan data sesi lokal.....	31
Gambar 3. 16 Gambar konfigurasi firebase	32
Gambar 3. 17 Gambar kode register dan penyimpana data akun di database	33
Gambar 3. 18 Gambar kode login.....	34
Gambar 3. 19 Gambar kode untuk update sesi	35
Gambar 3. 20 Gambar kode simpan sesi ke database.....	36
Gambar 3. 21 Gambar kode logout.....	37
Gambar 3. 22 Gambar kode unduh mushaf	40
Gambar 4. 1 Gambar Antarmuka login.....	42
Gambar 4. 2 Gambar email untuk reset password	43
Gambar 4. 3 Gambar antarmuka homepage	43
Gambar 4. 4 Gambar antarmuka register	44
Gambar 4. 5 Gambar notifikasi ketika pendaftaran sukses.....	45
Gambar 4. 6 Gambar halaman pengaturan ketika pengguna sudah login.....	46
Gambar 4. 7 Gambar notifikasi download.....	47

BAB I

PENDAHULUAN

1.1 Latar Belakang

Al-Qur'an adalah kitab suci yang terjaga dan tidak berubah. Melihat dari signifikannya, Al-Qur'an diajarkan dan dihafalkan oleh jutaan umat islam di dunia seperti pada awal diturunkan (Zakariah, Khan, Tayan, & Salah, 2017). Menghafal Al-Qur'an itu memiliki manfaat yang sangat baik untuk melatih memori otak (Aziz, Abdullah, Ahmad, Amat Mushim, & Shahrudin, 2019). Namun, jumlah penghafal Al-Qur'an saat ini masih relatif sedikit. Mengutip laporan Yasmina Hasni bahwa penghafal Al-Qur'an dari Arab Saudi hanya ada 6000 orang saja (Yuwanto, 2010). Warga Gaza memiliki sebanyak 10 ribu penghafal seluruh kitab suci Al-Qur'an setiap tahunnya, kendati jalur Gaza yang merupakan wilayah yang rawan konflik (Eksan, 2023). Jumlah penghafal Al-Qur'an di indonesia itu sendiri tergolong cukup rendah yakni 30.000 dari total sekitar 234 juta jiwa pada tahun 2010 (Yuwanto, 2010).

Menghafal Al-Qur'an sendiri bukanlah hal yang mudah, sederhana, dan dapat dilakukan setiap orang tanpa meluangkan waktu khusus, memberikan kesungguhan, dan juga keseriusan (Raghib, 2007). Adanya tantangan untuk menghafal yang disebabkan oleh kemampuan menghafal dan motivasi yang berbeda, yang dikarenakan penghafal bukan hanya fokus terhadap Al-Qur'an, namun juga pelajaran dan kegiatan lain (Ifana, 2023). Hal ini senada dengan pendapat pakar gangguan memori Zaldy Tan, MD yang diwawancarai oleh Cassie Tomlin, yang mana beliau mengatakan bahwa "melupakan sesuatu bukanlah hal yang aneh terutama jika seseorang sedang sibuk dan mudah untuk terganggu" (Tomlin, 2023). Hal ini juga diperburuk dengan memori manusia yang akan semakin menurun setiap bertambahnya usia (Susanto, Djojosoewarno, & Rosnaeni, 2009).

Berdasarkan uraian yang telah dipaparkan di atas, maka diperlukan sebuah alat atau aplikasi yang dapat menunjang kebutuhan penghafal, dalam hal ini terutama menunjang akses penghafal dan juga menyediakan sebuah pencatatan akan perkembangan hafalan penghafal. Oleh karena itu dibutuhkan sebuah penyimpanan berupa sesi yang disesuaikan atau dipersonalisasikan dengan kebutuhan penghafal. Sesi sendiri berfungsi untuk menjaga suatu

keadaan khusus milik pengguna, meliputi objek yang persisten dan identitas pengguna yang sudah terautentikasi (IBM, 2021).

Berdasarkan hasil pemaparan data dan fakta mengenai tantangan aplikasi untuk mendukung penghafal supaya dapat mencatat hafalan dan tidak lupa, maka penelitian ini bertujuan untuk mengembangkan sebuah aplikasi berbasis *mobile application* yang menasar kepada pengguna yang membutuhkan aplikasi Al-Qur'an dengan fitur sesi yang memiliki personalisasi sesuai dengan kebutuhan penghafal. Aplikasi ini diharapkan untuk dapat menyesuaikan kebutuhan para penghafal Al-Qur'an, dalam kasus ini adalah kebutuhan untuk mencatat perkembangan hafalan, supaya aplikasi dapat menunjang hafalan, dan dapat digunakan secara berkelanjutan.

1.2 Rumusan Masalah

- A. Bagaimana cara mengimplementasikan personalisasi dalam sebuah aplikasi Al-Qur'an berbasis mobile dengan menggunakan sesi untuk menyimpan data hafalan pengguna?.
- B. Apakah personalisasi dapat membantu memudahkan proses menghafal penghafal Al-Qur'an?.

1.3 Batasan Masalah

- A. Pengguna aplikasi tidak memiliki keterbatasan intelektual dan sudah terbiasa menggunakan gadget.
- B. Pengguna aplikasi sudah memasuki usia 15-35 tahun.

1.4 Tujuan Penelitian

- A. Membuat personalisasi progress menghafal melalui aplikasi Al-Qur'an berbasis *mobile* untuk memudahkan proses menghafal penghafal Al-Qur'an.

1.5 Manfaat Penelitian

- A. Memberikan kemudahan dalam menggunakan aplikasi.
- B. Memudahkan proses menghafal penghafal Al-Qur'an melalui aplikasi Al-Qur'an berbasis *mobile*.

1.6 Metodologi Penelitian

Metode yang digunakan dalam perancangan Aplikasi ini menggunakan metode *Software Development Life Cycle*(SDLC) yang terdiri dari:

- A. *Planning*, atau fase perencanaan pada fase ini akan dilakukan perencanaan dan pengumpulan data dari rencana yang sudah dibuat.
- B. Desain, pada fase ini pengolahan informasi-informasi yang telah didapatkan pada tahap analisis, setelah itu dimulailah proses perancangan. Pada langkah inilah perancangan aplikasi akan dibuat.
- C. Pengembangan, pada fase ini desain yang sudah dibuat akan diimplementasikan menjadi sebuah sistem sistem, dan jika sistem.
- D. Pengujian, pada fase ini desain yang sudah dikembangkan menjadi fitur-fitur aplikasi diujikan untuk memastikan hasil akhir dan juga memastikan umpan balik dari pengguna.

1.7 Sistematika Penulisan

Sistematika penulisan dalam penyusunan laporan tugas akhir ini terdiri dari beberapa bab, yang mencakup gambaran dari keseluruhan masalah dan penyelesaiannya. Berikut sistematika penulisan yang terbagi dalam 5 bab :

BAB I PENDAHULUAN

Pada bab ini berisi pembahasan latar belakang masalah, rumusan masalah, Batasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian dan sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Pada bab ini berisi pembahasan mengenai tinjauan terhadap penelitian yang pernah ada berhubungan dengan apa yang akan dirancang dan diimplementasikan serta teori dasar yang digunakan berhubungan dengan sistem untuk mengimplementasikan personalisasi.

BAB III METODOLOGI

Bab ini berisi uraian tentang analisis, desain dan personalisasi aplikasi quran menghafal.

BAB IV HASIL DAN PEMBAHASAN

Bab ini akan dibahas tentang pengimplementasian aplikasi serta membahas evaluasi pengujian aplikasi, baik dari segi isi konten maupun cara penggunaannya.

BAB V KESIMPULAN DAN SARAN

Bab ini merupakan bab terakhir yang akan membahas kesimpulan dan saran terhadap penelitian yang telah dilakukan pada tugas akhir.

BAB II

TINJAUAN PUSTAKA

Bab ini terdiri dari lima sub bab, Bagian pertama berisikan penelitian terdahulu yang terkait erat dengan masalah penelitian. Bagian kedua berisikan teori teori erat yang digunakan dalam penelitian. Bagian ketiga berisikan pengembangan aplikasi. Bagian keempat berisikan metode *Software Development Life Cycle(SDLC)* yang berisi perencanaan, desain, implementasi, dan pengujian. Bagian kelima berisikan *tool* pengembangan aplikasi mobile, pengujian sistem, pengujian fungsional, dan juga *User Acceptance Test (UAT)*.

2.1 Penelitian Terdahulu

A. Literatur pertama karya Jay P. Trivedi & Hemant Trivedi dengan judul “Investigating the Factors That Make a Fashion App Successful: The Moderating Role of Personalization”, Yang dipublikasikan pada tahun 2018 (Trivedi & Trivedi, 2018).

Penelitian ini memiliki membahas mengenai peran moderat personalisasi untuk meningkatkan aplikasi menjadi sukses. Hasil yang diperoleh dari penelitian ini menunjukkan bahwa ketiga dimensi kualitas yakni *Information Quality(INFQ)*, *System Quality(SYSQ)*, dan *Service Quality(SERQ)* mempengaruhi *Application Satisfaction(APS)* secara signifikan. Juga, APS menunjukkan pengaruh yang signifikan terhadap *Purchase intentions(PI)*. *Personalization(PER)* diamati untuk memoderasi hubungan antara ketiga kualitas dimensi dan APS. PER meningkatkan kekuatan hubungan antara variabel independen INFQ, SYSQ, SERQ dan dependen variabel APS sehingga memoderasi hubungan. Dilihat dari hasil ini, mode pemasar dapat menggunakan data-led strategi yang memungkinkan aplikasi untuk belajar dari profil pengguna dan memberikan informasi, layanan, dan kinerja sistem yang dipersonalisasi, untuk mendapatkan tingkat kepuasan pengguna yang lebih tinggi, yang dapat mengarah ke PI.

Penelitian ini memiliki keunggulan karena menjabarkan faktor-faktor yang dapat dipersonalisasi untuk meningkatkan kepuasan pengguna dan keinginan pengguna untuk melakukan pembelian, namun penelitian ini masih memiliki kekurangan berupa data yang masih terbatas pada pengguna dengan usia 21-25 tahun yang tinggal di kota metropolitan

India. Personalisasi pada aplikasi ini masih memiliki ruang untuk dikembangkan pada aplikasi lain kendati melihat dari respon positif dari pengguna aplikasi ini sendiri.

Pada penelitian ini menunjukkan bahwa personalisasi memiliki faktor yang dapat meningkatkan kepuasan pengguna. Penelitian ini juga menunjukkan bahwa personalisasi meningkatkan keinginan pengguna untuk memakai fitur aplikasi. Personalisasi pada penelitian ini berupa penyesuaian aplikasi berdasarkan pengguna memiliki peran menjembatani antara kepuasan pengguna dan kualitas aplikasi, karena aplikasi ini sudah disesuaikan dengan kebutuhan pengguna, sehingga terlihat timbal balik antara kepuasan pengguna dan kualitas aplikasi di mata pengguna.

B. Literatur kedua karya Natalia Kucirkova dengan judul “Children’s agency by design: Design parameters for personalization in story-making apps”, yang dipublikasikan pada tahun 2019 (Kucirkova, 2019).

Penelitian ini membahas mengenai parameter desain untuk personalisasi di *story-making apps*. Penelitian ini mendapati bahwa tingkat personalisasi yang tergabung pada produk akhir dan bagaimana cara hal tersebut menyediakan ruang untuk agensi anak-anak itu penting secara teori dan pendidikan dengan mempertimbangkan desain multimedia. Penelitian ini mensintesis penelitian internasional dengan aplikasi OS dengan maksud untuk mengembangkan desain yang secara strategis menggunakan personalisasi dan memahami hubungan antara personalisasi, kustomisasi, dan individualisasi.

Penelitian ini memiliki keunggulan dengan menyediakan framework desain strategis untuk personalisasi dan memahami hubungan personalisasi, kustomisasi, dan individualisasi pada *story-making apps*, namun penelitian ini masih memiliki kekurangan seperti konsep personalisasi yang masih terbatas pada personalisasi di bidang pendidikan.

pada penelitian ini Juga menunjukkan bahwa personalisasi dapat berkolaborasi dengan aspek lain. Aspek lain yang terlihat pada penelitian ini adalah kustomisasi, dan individualisasi. Sehingga membuka peluang personalisasi untuk dikolaborasikan dengan aspek lain.

C. Literatur ketiga Nasim Almadri et al dengan judul “A Real-Time Personalized Noise Reduction Smartphone App for Hearing Enhancements”, yang dipublikasikan pada tahun 2018 (Alamdari, Yaraganalu, & Kehtarnavaz,, 2018).

Penelitian ini membahas aplikasi *real-time* yang dipersonalisasi untuk mengurangi suara dan meningkatkan daya dengar. Penelitian ini memiliki hasil dari 2 data uji yang diambil dari 2 publik domain dataset, dataset yang pertama berisikan suara pembicaraan yang bersih, dan yang kedua berisikan suara ribut dari lingkungan sekitar. Kedua dataset tersebut kemudian diujikan menggunakan *no noise reduction* (NNR), dan *fixed noise reduction* (FNR) dan menunjukkan bahwa aplikasi *personalized noise reduction* (PNR) memiliki konsistensi yang lebih tinggi ketimbang aplikasi lainnya.

Penelitian ini memiliki kelebihan yakni aplikasi yang dipersonalisasi ini memiliki performa yang relatif sama seperti aplikasi pada umumnya, namun penelitian ini memiliki kelemahan yakni adanya perbedaan penggunaan performa pada IOS dan Android.

Pada penelitian ini menunjukkan bahwa aplikasi yang menggunakan personalisasi dalam kasus ini adalah untuk pengurangan suara ribut lebih memiliki konsistensi untuk orang-orang yang membutuhkan.

D. Literatur keempat Tan, C. W., Ling, L., Yu, P., Hang, C. N., & Wong, M. dengan judul “Mathematics Gamification in *Mobile App Software for Personalized Learning at Scale*” (Tan, Ling, Yu, Hang, & Wong, 2020).

Penelitian ini membahas mengenai gamifikasi matematika dengan sebuah aplikasi yang dipersonalisasi sesuai dengan pengguna. Aplikasi yang dipersonalisasi tadi kemudian diberikan kepada responden untuk mereka coba. Dari penelitian ini ditemukan hasil jika aplikasi ini dapat digunakan pengguna untuk belajar sesuai kemampuan mereka sendiri, karena sudah memiliki personalisasi. Namun, aplikasi ini masih memiliki kekurangan berupa personalisasi hanya terpaut pada konsep gamifikasi dan matematika. Sehingga masih dapat dilakukan eksplorasi terhadap personalisasi pada konsep lainnya,

pada penelitian ini didapati bahwa pelajaran yang dipersonalisasi memiliki sebuah dampak yang jelas. Dampak tersebut dapat dilihat dari responden yang dapat menyesuaikan gaya belajar, sesuai dengan kemampuan.

Dalam penelitian terdahulu posisi personalisasi, masih bersifat variatif. Namun, jika dilihat kembali penelitian-penelitian terdahulu tersebut menggunakan personalisasi, yakni penyesuaian aplikasi maupun fitur-fitur dengan kebutuhan pengguna atau responden penelitian.

E. Literatur kelima Ehrler, F., Lovis, C., & Blondon, K dengan judul “A mobile phone app for bedside Nursing care: design and development using an adapted software development life cycle model” (Ehrler, 2019).

Penelitian ini membahas mengenai pengembangan sebuah aplikasi berbasis *mobile*, yang didalamnya terdapat sebuah personalisasi menggunakan *machine learning*. Penelitian ini menggunakan metode dan desain Software Development Life Cycle(SDLC), yang kemudian disesuaikan dengan kebutuhan pengembangan software. Penelitian ini memiliki tujuan untuk menunjukkan bahwa SDLC yang digunakan berdasarkan bukti dan timbal-balik dari pengguna dapat menjalankan proses desain dan pengembangan. Penelitian ini memiliki hasil yang menunjukkan bahwa SDLC memiliki kekuatan dari keterkaitan pengguna dan juga ekologi pada fase evaluasi. Namun penggunaan ini masih terbataas di bidang Kesehatan.

Dalam penelitian terdahulu tersebut dapat diketahui bahwa SDLC dapat disesuaikan dengan kebutuhan. SDLC juga dapat melibatkan pengguna dalam proses pengembangan dan juga evaluasi atau pengujian.

F. Literatur keenam Muhammad Ulil Albab Surya Negara dengan judul “Digitalisasi Mushaf Penghafal Al-Qur'an untuk Memudahkan Mengafal Al-Qur'an” (Negara, 2022).

Penelitian ini membahas mengenai pengembangan sebuah aplikasi Al-Qur'an berbasis *mobile*, yang bertujuan untuk memudahkan menghafal Al-Qur'an. Dengan memberikan fitur-fitur seperti masukkan data capaian menghafal, membuka mushaf ,menavigasi letak ayat, mencari surat, menandai ayat yang telah dihafalkan, menutup surat, mengetahui progress hafalan, melihat terjemahan, dan mendengarkan bacaan. Namun hasil penelitian masih belum mempunyai personalisasi untuk penghafal supaya penghafal dapat mengakses data progress hafalan mereka pada perangkat lain, yang membutuhkan adanya sebuah

penyimpanan progress yang terdapat pada *database*, serta mushaf pada aplikasi hanya terdapat juz 30 saja untuk.

Posisi penelitian ini terhadap penelitian di atas adalah melanjutkan dengan menambahkan fitur-fitur yang belum terdapat pada penelitian sebelumnya. Seperti fitur autentikasi untuk pengguna, penyimpanan data menggunakan sesi, penghapusan data lokal pengguna jika pengguna *logout*, dan download mushaf.

Posisi penelitian ini terhadap penelitian-penelitian terdahulu adalah menggunakan personalisasi dengan dimensi penggunaan yang berbeda, yakni penggunaan dimensi personalisasi pada penelitian ini terletak pada dimensi pendidikan dan dimensi kepuasan pengguna. Dimensi tersebut dapat diperoleh dengan mengkolaborasikan personalisasi dengan sesi untuk menyimpan data hafalan pengguna, karena pengguna membutuhkan sebuah aplikasi yang memiliki fitur sesuai dengan keinginan mereka.

2.2 Kajian Teori

2.2.1 Quran Digital

Saat ini terdapat banyak sekali aplikasi Al-Qur'an digital yang tersedia pada platform mobile, seperti Quran digital yang dapat digunakan secara offline, dan bahkan ada sebuah perusahaan yang sudah membuat prototipe quran digital yang dapat mengajarkan cara membaca Al-Qur'an (Zakariah, Khan, Tayan, & Salah, 2017). Berdasarkan survei dari Muhammad Khurram Khan dan Yasser M.Alginahi (Khurram Khan & M. Alginahi, 2013) didapatkan hasil yang menunjukkan sebesar 30% responden lebih menyukai Al-Qur'an digital yang dapat di download melalui toko penyedia aplikasi di Android, IOS dan lainnya. Jika dilihat dari kedua hal tersebut dapat kita ketahui bahwa Al-Qur'an digital yang tersedia pada *platform* tertentu merupakan aplikasi yang menyediakan Al-Qur'an untuk dibaca maupun digunakan oleh pengguna, yang dapat diperoleh dengan cara mengunduh aplikasi tersebut baik dari toko penyedia aplikasi maupun sumber lainnya.

2.2.2 Personalisasi

Menurut (Optimizely, n.d.) personalisasi adalah sebuah tindakan untuk menampilkan konten, produk, ataupun sebuah poin digital yang disesuaikan untuk sejumlah pengguna

dengan tujuan untuk meningkatkan relevansi. Jadi personalisasi dapat diartikan sebagai cara seorang developer untuk menyesuaikan aplikasi agar relevan dengan kebutuhan target pengguna. Tujuan utama dari personalisasi adalah untuk memberikan konten dan fungsionalitas yang sesuai dengan kebutuhan atau ketertarikan pengguna secara spesifik, tanpa memerlukan usaha dari target pengguna (Nielsen Norman Group, n.d.).

2.2.3 Sesi

Sesi yang dikenal juga sebagai “*visit*”, “*journey*”, atau “*clickpath*” adalah serangkaian aksi pengguna yang mereka lakukan di dalam aplikasi pada saat tertentu (Dynatrace, 2024). Secara khusus sesi merujuk kepada rangkaian kegiatan yang dilakukan untuk memenuhi kebutuhan informasi, tanpa memperhatikan berapa lama waktu yang dibutuhkan dan apakah ada jeda di tengahnya (Mehrzadi & Feitelson, 2012). Rangkaian kegiatan yang telah disebutkan sebelumnya merupakan rangkaian kegiatan pengguna dalam mengoperasikan sebuah perangkat atau aplikasi. Tujuan dari sesi adalah menjaga data pengguna sepanjang interaksi mereka dengan aplikasi (Fatah, 2023). Sehingga data tidak sewaktu-waktu hilang jika sesi.

2.3 Pengembangan Aplikasi

Terdapat beberapa pengembangan aplikasi yang ada saat ini diantaranya adalah pengembangan aplikasi berbasis mobile, pengembangan aplikasi berbasis web, dan pengembangan aplikasi berbasis desktop atau komputer. Jika dilihat dari kebutuhan dalam pengembangan aplikasi, maka perlu diputuskan pengembangan aplikasi berbasis apakah yang akan digunakan.

Pengembangan aplikasi berbasis mobile itu sendiri akan menghasilkan sebuah aplikasi. Aplikasi mobile itu sendiri mudah, ramah pengguna, tidak mahal, dapat diunduh dan dijalankan di berbagai macam perangkat mobile termasuk perangkat yang mahal ataupun yang berada pada tingkatan entri (Islam & Mazumder, 2010). Sehingga dapat disimpulkan bahwa perangkat mobile memiliki akses yang lebih mudah di dapat di berbagai kalangan.

Dilihat dari kemudahan akses dan tujuan dari penelitian maka saya selaku peneliti menggunakan pengembangan aplikasi berbasis mobile sebagai pilihan dari jenis pengembangan aplikasi yang akan dikembangkan.

2.4 Software Development Life Cycle(SDLC)

Software Development Life Cycle(SDLC) merupakan sebuah *framework* yang mendefinisikan proses, pekerjaan, dan aktivitas yang berkaitan dengan pengembangan aplikasi, yang memiliki tujuan utama untuk memenuhi keinginan pengguna (H, 2023). SDLC sendiri memiliki keunggulan untuk dapat menyelesaikan *software* dalam tenggat waktu tertentu dan tetap menjaga kualitas produk sesuai dengan standar (Mishra, 2013). Keunggulan tersebut senada dengan pendapat (AWS, n.d.) yang mengatakan bahwa SDLC memiliki sistematika yang baik dan kepuasan pengguna yang lebih baik. Menurut (AWS, n.d.), SDLC terbagi menjadi beberapa tahap sebagai berikut.

2.4.1 Perencanaan

Pada fase perencanaan biasanya meliputi tugas-tugas seperti analisis, penjadwalan, estimasi pengeluaran dan alokasi. Pengembang kemudian mengumpulkan kebutuhan-kebutuhan dari beberapa *stakeholder*, seperti pengguna, pakar *internal* maupun *external*, dan manajer untuk membuat dokumen kebutuhan.

2.4.2 Desain

Pada fase desain, pengembang menganalisa kebutuhan dan mengidentifikasi solusi terbaik untuk mengembangkan perangkat lunak. Sebagai contoh, seperti mengintegrasikan modul yang sudah pernah dibuat, membuat keputusan untuk teknologi yang akan digunakan, dan alat-alat pengembangan.

2.4.3 Implementasi

Pada fase implementasi, pengembang akan membuat kode untuk kebutuhan produk. Pengembang akan menganalisa kebutuhan untuk mengidentifikasi tugas tugas yang akan dilakukan secara harian.

2.4.4 Pengujian

Pengembang akan melakukan pengujian untuk melihat apakah ada kesalahan atau *bug* yang terdapat pada perangkat lunak. Analisa kualitas yang meliputi pengujian perangkat lunak dari error dan pengujian apakah sudah memenuhi kebutuhan pengguna.

2.5 Pengembangan Aplikasi mobile

2.5.1 Alat-alat pengembangan aplikasi mobile

Idealnya pengembangan aplikasi mobile memiliki tujuan untuk mengembangkan aplikasi menggunakan bahasa pemrograman yang terdukung dan terdokumentasi dengan baik beserta sekumpulan API dan aplikasi dapat tersedia di seluruh perangkat mobile terlepas dari jenis perangkat dan jaringan (Smutny, 2012). Alat-Alat yang harus satu diantaranya adalah bahasa pemrograman. Terdapat banyak bahasa pemrograman dan framework yang mendukung pengembangan aplikasi berbasis mobile diantaranya adalah Kotlin, Flutter, Java, dan React Native.

Kotlin adalah bahasa pemrograman bertipe statis yang berjalan di Java Virtual Machine(JVM) dan juga dapat dikompilasikan ke JavaScript sebagai *source code* (Bose, 2018). Nilai jual utama Kotlin dibandingkan dengan bahasa pemrograman seperti Java adalah kemampuan Kotlin untuk mengurangi *error* melalui kumpulan konstruktor yang baru (Ardito, Coppola, Malnati, & Torchiano, 2020). Kesalahan-kesalahan biasa yang sering terjadi dikarenakan desain dapat dicegah dengan menggunakan Kotlin, sehingga menghasilkan lebih sedikit kegagalan sistem dan *crash*-nya aplikasi (Bose, 2018). Namun, Kotlin juga memiliki kekurangan seperti masih kurangnya jumlah dokumentasi dibandingkan bahasa pemrograman lainnya (Bose, 2018). Kotlin juga hanya dapat digunakan untuk mengembangkan aplikasi yang menggunakan Android.

Flutter adalah *cross-platform framework* yang memiliki tujuan untuk mengembangkan aplikasi dengan performa yang tinggi (Wu, 2018). Flutter sendiri merupakan UI *toolkit* milik Google yang ditujukan untuk mengembangkan aplikasi mobile, website dan desktop yang dikompilasikan secara *native* dan ditulis menggunakan Dart (Eshetu Fentaw, 2020). Flutter memiliki kelebihan seperti tampilan dan properti *Stateless Widget* yang tidak akan pernah berubah sepanjang umur mereka, memiliki properti *immutable*, dan memiliki *state* masing-masing (Eshetu Fentaw, 2020). Flutter sendiri memiliki kelemahan seperti menggunakan Main View Controller(MVC), yang mana MVC mengizinkan *view* untuk berkomunikasi secara langsung dengan *controller* dan *model*. Sifat MVC ini dapat merusak konsistensi data dan meningkatkan kebingungan (Wu, 2018).

Java adalah bahasa pemrograman yang dapat ditemukan di berbagai jenis perangkat dari ponsel pintar, hingga komputer mainframe (Bose, S., 2018). Java sendiri sering dikatakan kurang efisien (Prechelt, 2000).

React Native adalah *open-source cross-platform JavaScript framework* yang dibuat untuk menangani dilema perbedaan platform yakni Android dan IOS (Wu, 2018). React Native sendiri menggunakan styling CSS karena mendapatkan pengaruh besar dari framework React. React Native sendiri memiliki komunitas yang cukup besar dan lebih makmur, sebagai contoh React Native memiliki library yang lebih matang. Namun React Native memiliki kekurangan navigasi yang belum modern dan memiliki pengalaman selayaknya aplikasi *native* (Wu, 2018).

2.5.2 React Native

Tujuan utama dari React Native itu sederhana, yaitu seorang pengembang tidak perlu pengetahuan atau perlu untuk menghabiskan waktu yang sangat banyak untuk membuat sebuah aplikasi mobile, karena setidaknya ada dua aplikasi yang perlu dikembangkan untuk mendukung IOS dan Android (Danielsson, 2016). Sehingga aplikasi yang dikembangkan menggunakan React Native memiliki kebebasan yang lebih, dibandingkan dengan bahasa pengembangan *native*, karena dapat dikembangkan untuk dua ekosistem Android dan IOS.

2.5.3 Firebase

Firebase merupakan sebuah set perangkat pengembangan *cloud-based* yang membantu dalam pengembangan aplikasi berbasis mobile (Hanna & Rosencrance, 2023). Firebase sendiri menawarkan fitur-fitur seperti autentikasi pengguna, penyimpanan, dan juga *hosting* untuk aplikasi (Adalyn, 2022).

2.5.4 Unified Modelling Language (UML)

Unified modelling language(UML), merupakan suatu alat untuk menggambarkan dan mendokumentasikan hasil dari desain dan Analisa yang berisi sintak dalam memodelkan sistem secara visual (Haviluddin, 2016). Menurut Lucidchart terdapat beberapa jenis diagram UML seperti *Use case diagram*, *Sequence diagram*, *Activity diagram*.

2.5.5 Pengujian Sistem

Motivasi utama dalam melakukan pengujian pada proses pengembangan adalah untuk mengurangi risiko terhadap biaya, atau lebih parah kegagalan proyek yang tidak terencana (Everett & McLeod, 2007).

2.5.6 *Blackbox testing*

Black box testing yang juga dikenal sebagai *Behavioral Testing* adalah pengujian yang dilakukan untuk mengamati hasil *input* dan *output* dari perangkat lunak tanpa harus mengetahui struktur dari kode dalam perangkat lunak (Setiawan, 2021). *Black box testing* memiliki 2 tipe yakni pengujian fungsionalitas dan non-fungsionalitas, tujuan pengujian fungsionalitas adalah untuk memastikan fungsi aplikasi sudah sesuai dengan *input* dan mencocokkan output dengan kebutuhan (Geeksforgeeks, 2023).

2.5.7 *User Acceptance Testing (UAT)*

User Acceptance Testing (UAT) adalah tahap *testing* terakhir dan terpenting dari tahapan-tahapan *testing* yang umum dilakukan (Andriana, 2020). UAT sendiri adalah sebuah pengujian yang digunakan untuk memastikan bahwa aplikasi telah memenuhi poin-poin dalam dokumen serta mampu untuk diterima oleh pengguna (Meilinaeka, 2023). Pengujian UAT lebih berfokus kepada sisi bisnis untuk menguji apakah sistem telah memenuhi kebutuhan dengan skenario uji dan data yang mewakili pengguna aktual. Menurut (Rice, n.d.) Untuk mendapatkan *Acceptance Criteria Coverage* yang berfungsi untuk mengukur tingkat kepuasan pengguna dapat dengan menggunakan rumus berikut (2.1)

$$ACC = \frac{\text{Jumlah Bobot Penilaian}}{\text{Jumlah Bobot Total}} \cdot 100 \quad 2.1$$

Penilaian dapat didapatkan dari pengujian kepada pengguna dengan memberikan pertanyaan dan bobot jawaban dari masing masing pertanyaan. Menurut (Rice, n.d.) data ACC seharusnya mudah untuk didapat dan ACC harus mendekati angka 100%.

UAT tidak seperti pengujian-pengujian lainnya. Menurut (Bugwolf, 2016) elemen penting pada UAT yang harus diperhatikan berkaitan dengan pengguna adalah *business requirements, business processes, dan user expectations*.

BAB III METODOLOGI

Aplikasi ini akan dikembangkan menggunakan metode *Software Development Life cycle* (SDLC). Dalam mengembangkan aplikasi menggunakan SDLC menurut AWS Amazon pertama adalah melakukan perencanaan, kedua melakukan desain, ketiga melakukan implementasi, dan yang keempat melakukan pengujian.

3.1 Perencanaan

Pada fase perencanaan ini untuk mendukung pengembangan aplikasi maka dilakukan analisis kebutuhan dan perencanaan alur proses sistem.

3.1.1 Analisis Kebutuhan

Berdasarkan pengambilan data dari beberapa *stakeholder* dan juga penghafal maka didapatkan beberapa kebutuhan seperti pada tabel 3.1.

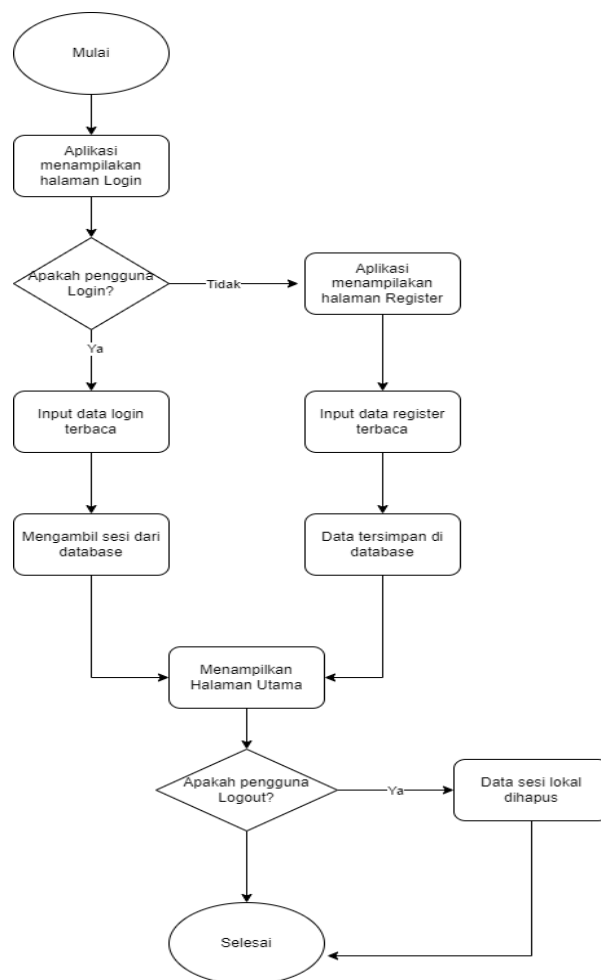
Tabel 3. 1 Tabel kebutuhan

NO.	Kebutuhan	
	Fungsional	Non-Fungsional
1.	Aplikasi mampu membuat pengguna untuk memilih berdasarkan keinginannya untuk <i>login</i> ataupun tidak.	Aplikasi dikembangkan dengan berbasis <i>mobile</i>
2.	Aplikasi mampu untuk menyimpan data sesi pengguna .	Aplikasi menggunakan <i>database</i> noSQL Firebase
3.	Aplikasi mampu untuk mengembalikan data sesi pengguna ketika <i>login</i> .	Aplikasi menggunakan <i>cross-platform framework</i> React Native.

4.	Aplikasi mampu menghapus data sesi pengguna ketika pengguna <i>logout</i> .	
----	---	--

3.1.2 Perencanaan Alur Proses Sistem

Setelah kebutuhan-kebutuhan teridentifikasi. Tahap berikutnya adalah membuat perencanaan alur proses sistem berdasarkan kebutuhan yang telah berhasil diidentifikasi.



Gambar 3. 1 Gambar alur proses sistem

Berdasarkan gambar 3.1, aplikasi awalnya akan menampilkan halaman *login* yang mana pengguna akan diberikan pilihan untuk melakukan *login* ataupun *register*. Ketika pengguna melakukan *login*, sistem akan membaca dan mengecek data login tersebut. Setelah dikonfirmasi sistem akan mengambil data sesi pengguna dari *database*, dan mengarahkan pengguna ke halaman utama. Kemudian, jika pengguna tidak melakukan *login*, maka

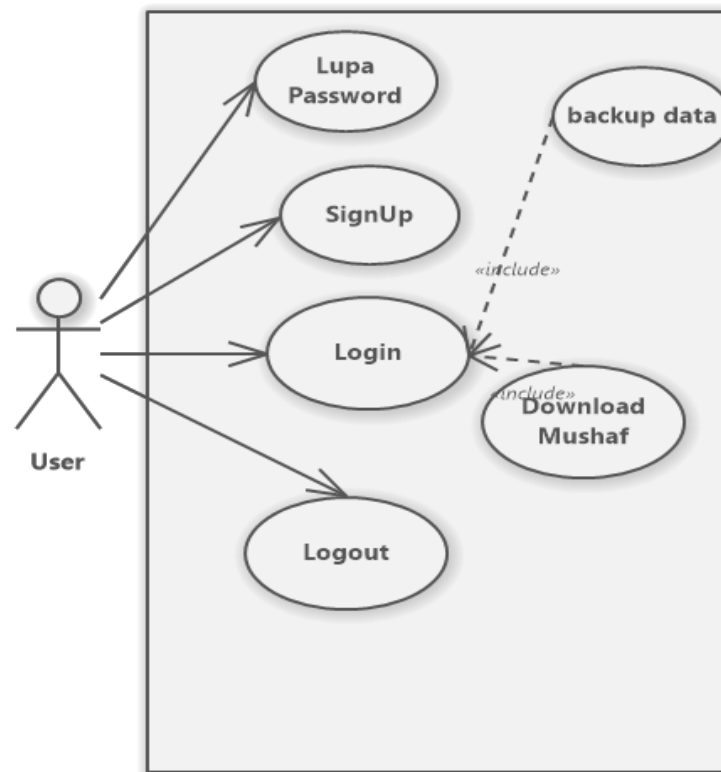
pengguna menekan tombol register dan akan dialihkan ke halaman *register*, Setelah pengguna mengisi formulir registrasi data akan terbaca oleh sistem dan data akan tersimpan di *database*. Sama halnya seperti saat *login* sukses, ketika pengguna berhasil melakukan registrasi pengguna akan diarahkan ke halaman utama. Jika pengguna melakukan *logout*, maka data sesi lokal pengguna akan terhapus.

3.2 Desain

Setelah melakukan tahanan sebelumnya (perencanaan). Aplikasi akan melewati tahap desain yang pertama adalah desain *usecase diagram*, kedua desain *login*, ketiga desain *register*, keempat desain *logout*, kelima desain *database*, dan keenam desain *download*.

3.2.1 Desain Usecase Diagram

Berdasarkan data yang diperoleh dari tahap sebelumnya (perencanaan). Maka dibuatlah sebuah *use case* untuk mencakup kebutuhan kebutuhan pada tahap sebelumnya.

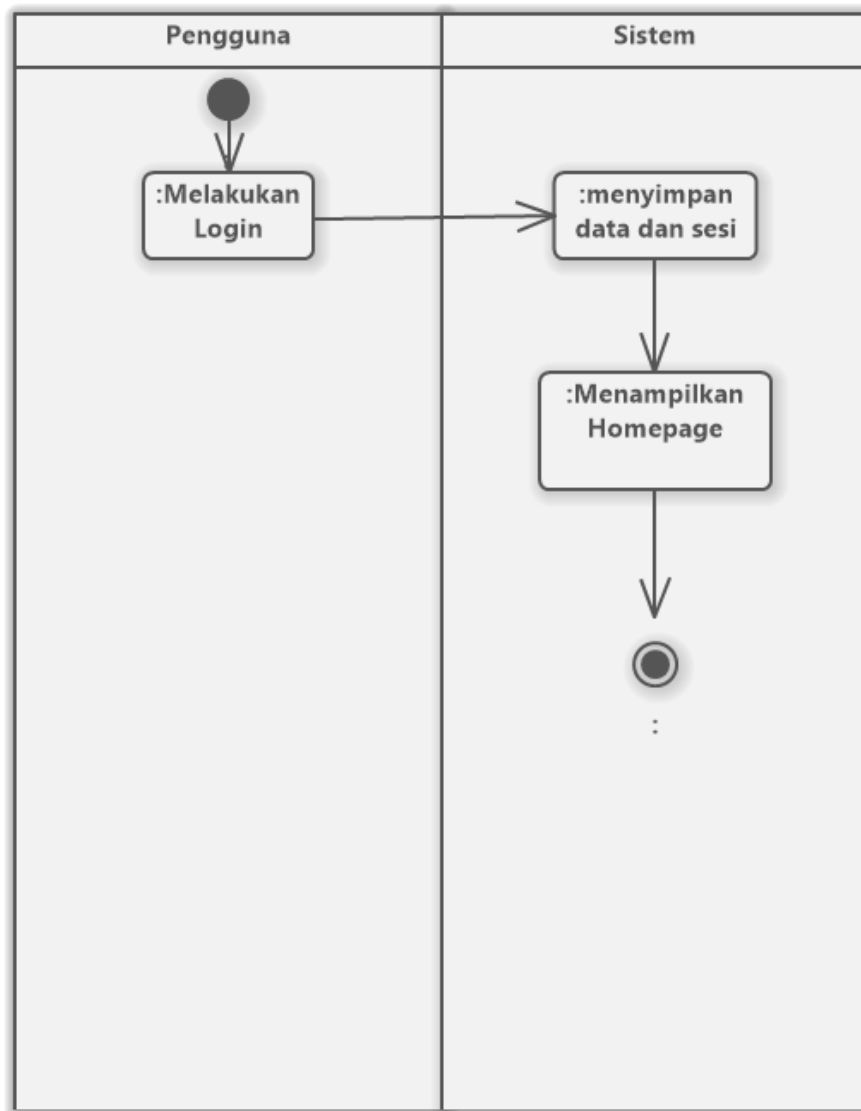


Gambar 3. 2 Gambar usecase pengguna

Gambar 3.2 menunjukkan *usecase diagram* pengguna. diagram ini merupakan hasil dari perancangan alur proses sistem dan juga kebutuhan-kebutuhan yang sudah didapatkan. Pada gambar di atas dapat dilihat bahwa terdapat beberapa fitur yang dapat diakses pengguna. Fitur-Fitur tersebut meliputi lupa password, *signup*, *login*, masuk sebagai tamu, *download*, dan juga *logout*.

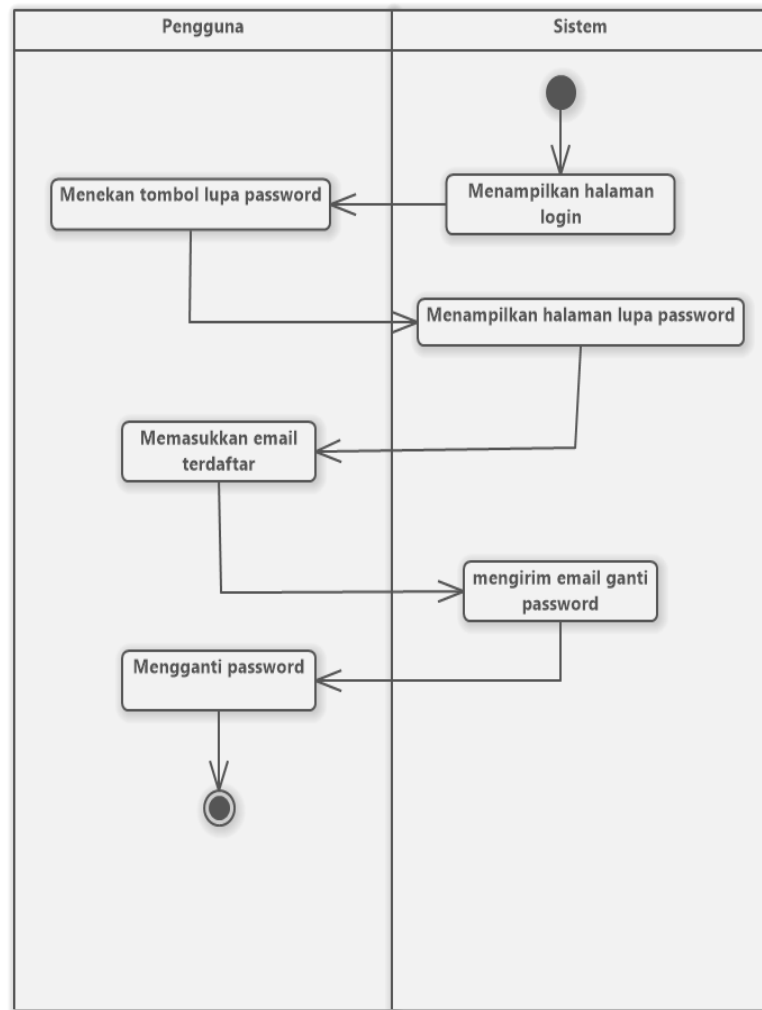
3.2.2 Desain Login

Setelah *use case* didapatkan mulai dilakukan pembuatan activity diagram untuk *login*.



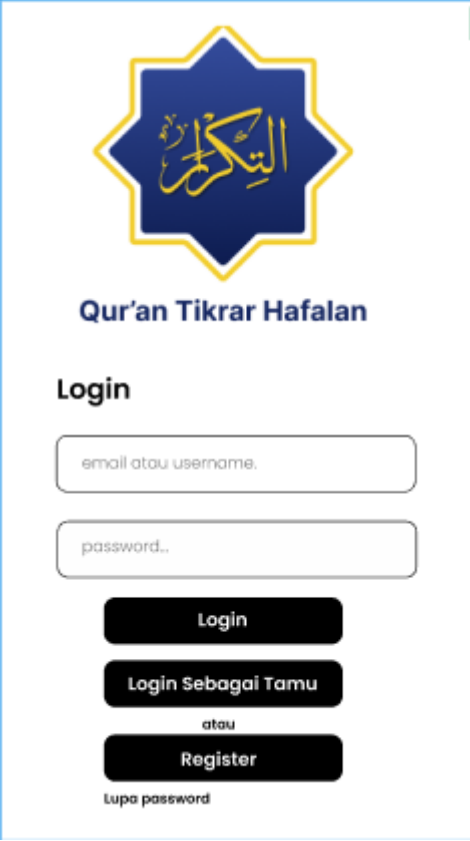
Gambar 3.3 Gambar activity diagram login

Gambar 3.3 merupakan *activity diagram* pengguna yang melakukan *login*. Dapat dilihat bahwa ketika pengguna melakukan *login* maka sistem akan mengambil data sesi pengguna dan mengarahkan pengguna ke halaman utama.



Gambar 3. 4 Gambar activity diagram lupa password

Gambar 3.4 merupakan *activity diagram* pengguna yang menekan tombol lupa *password* pada halaman *login*. Dapat dilihat bahwa ketika pengguna menekan tombol lupa *password* maka sistem akan menampilkan halaman lupa *password* yang berisikan sebuah *input* yang mana pada bagian itu pengguna dapat memasukkan *email* terdaftar mereka untuk mendapatkan *email* dari sistem untuk mereset kata sandi pengguna. Kemudian pengguna dapat memasukkan kata sandi atau *password* baru.



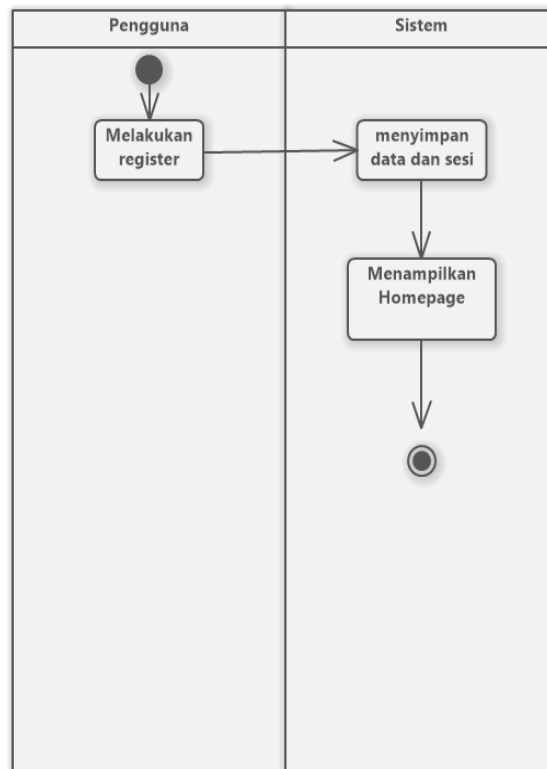
The image shows a prototype of a login page. At the top center is a blue star-shaped logo with yellow Arabic calligraphy. Below the logo is the title "Qur'an Tikrar Hafalan" in bold blue text. Underneath is the heading "Login" in bold black text. There are two input fields: the first is labeled "email atau username." and the second is labeled "password..". Below these fields are three buttons: "Login", "Login Sebagai Tamu", and "Register". At the bottom left, there is a link labeled "Lupa password".

Gambar 3. 5 Gambar prototype figma halaman login

Gambar 3.5 merupakan gambar desain prototipe dari halaman *login*. Pada desain ini terdapat kebutuhan yang harus dimasukkan pengguna yang telah mendaftar untuk dapat melakukan login. Selain itu terdapat opsi login sebagai tamu, *register* untuk melakukan registrasi, dan juga lupa password.

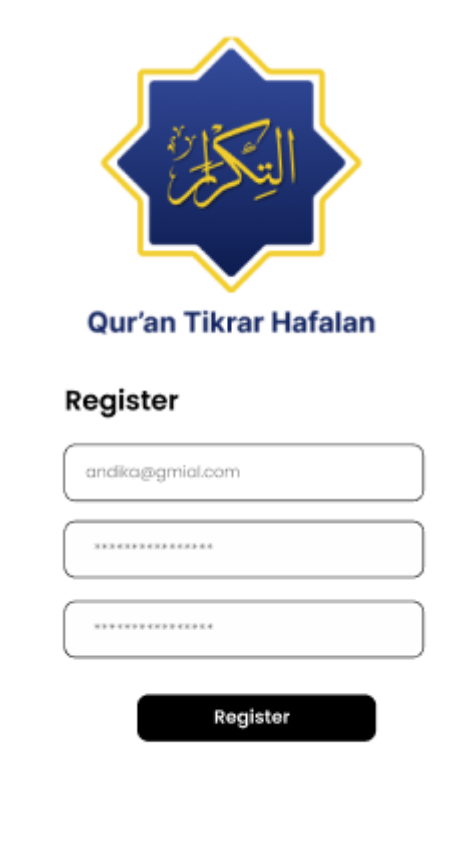
3.2.3 Desain Register

Kemudian melanjutkan pembuatan *activity diagram* untuk *register*.



Gambar 3. 6 Gambar activity daigram register

Gambar 3.6 merupakan *activity diagram* pengguna yang melakukan *register*. Dapat dilihat bahwa ketika pengguna melakukan *register* maka sistem akan menyimpan data sesi pengguna dan mengarahkan pengguna ke halaman utama.



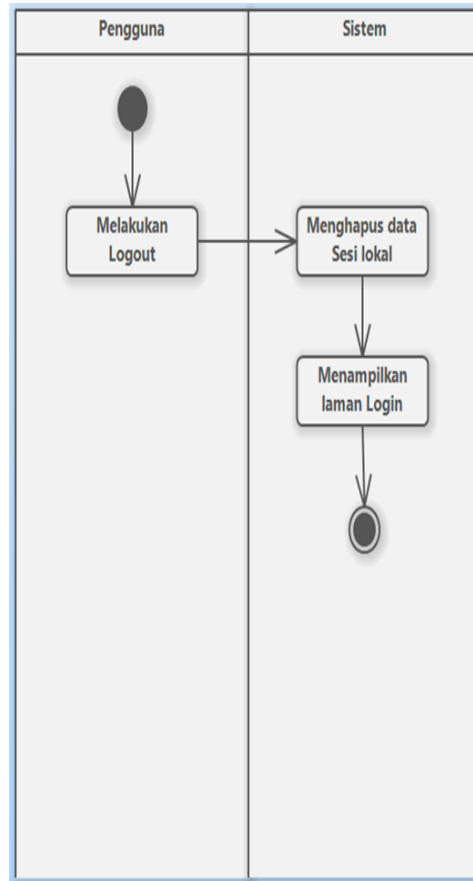
The image shows a Figma prototype of a registration page. At the top center is a blue star-shaped logo with a yellow outline, containing the Arabic word 'التكرار' (Al-Tikrar) in yellow. Below the logo is the text 'Qur'an TIKRAR Hafalan' in a bold, dark blue font. Underneath this is the heading 'Register' in a bold, black font. The registration form consists of three input fields: the first contains the email address 'andika@gmail.com', the second and third are masked with asterisks. Below the input fields is a black button with the text 'Register' in white.

Gambar 3. 7 Gambar prototype figma halaman register

Gambar 3.7 merupakan gambar desain prototipe halaman *register*. Desain prototipe ini menunjukkan kebutuhan pengguna untuk dapat melakukan registrasi akun. Kebutuhan-kebutuhan tersebut yakni *email*, dan *password*.

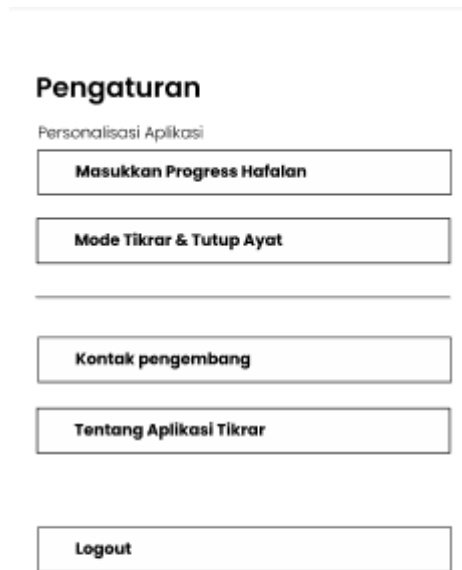
3.2.4 Desain Logout

Dilanjutkan dengan pembuatan *activity diagram logout*.



Gambar 3. 8 Gambar activity diagram logout

Gambar 3.8 merupakan *activity diagram* pengguna yang melakukan *logout*. Dapat dilihat bahwa ketika pengguna melakukan *logout* maka sistem akan menghapus data sesi pengguna pada penyimpanan lokal dan mengarahkan pengguna ke halaman *login*.

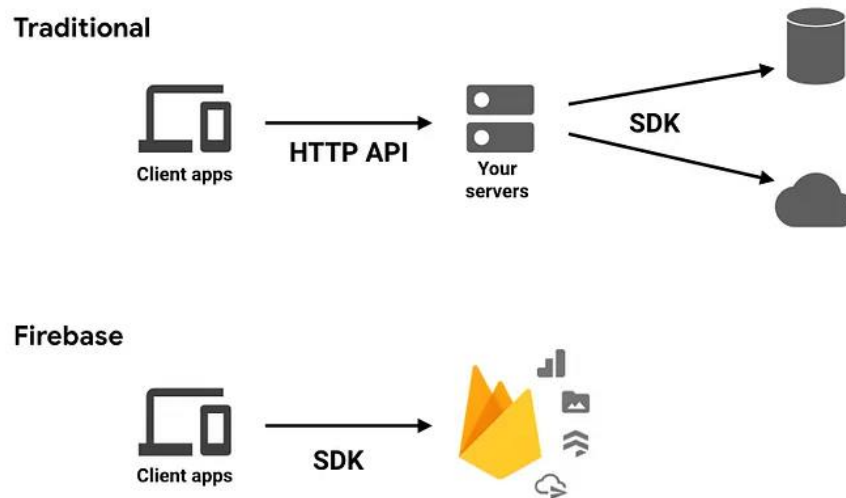


Gambar 3. 9 Gambar prototype figma halaman pengaturan

Gambar 3.9 merupakan gambar desain prototipe halaman pengaturan. Pada halaman ini terdapat fitur *logout* yang dapat digunakan pengguna untuk logout dari *aplikasi*.

3.2.5 Desain Database

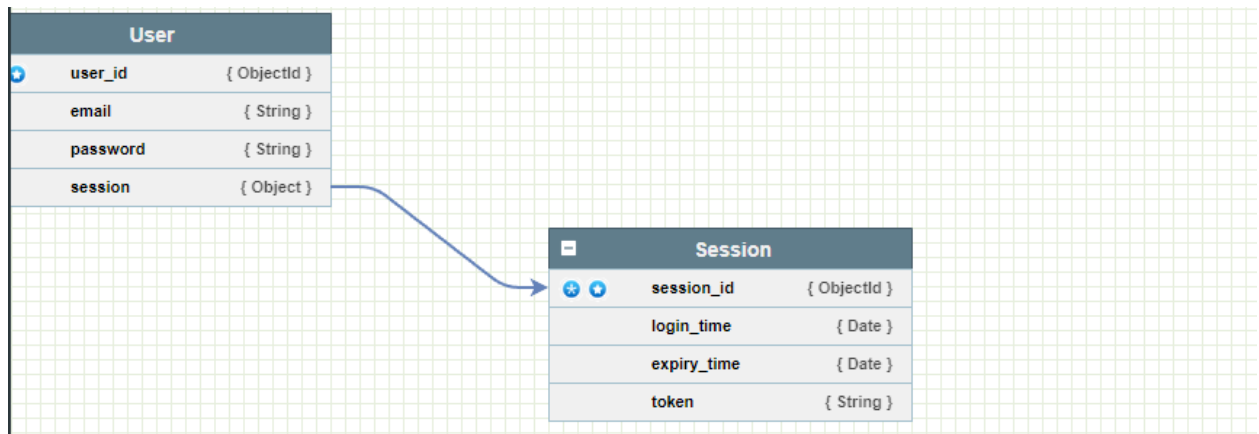
Setelah kebutuhan pada tahap sebelumnya(desain) terkumpul. Maka dipilihlah *database* untuk sistem dan juga pembuatan relasinya.



Gambar 3. 10 Gambar arsitektur firebase

Sumber:

Gambar 3.10 arsitektur *database Firebase*. *Firebase* sendiri merupakan *noSQL database*. *NoSQL database* itu sendiri banyak digunakan bahkan hampir digunakan di setiap industri (MongoDB, n.d.). Jika dilihat dari gambar di atas *Firebase* menyediakan arsitektur yang lebih sederhana, karena di dalam *firebase* sendiri terdapat fitur-fitur pendukung yang sudah terintegrasi. Seperti *Firebase Cloud Messaging* (FCM) yang memiliki fungsi untuk mengirimkan notifikasi, *Firebase Firestore* yang dapat menampung data-data, *Firebase Real-Time Database* yang dapat mengolah data secara *real time*, dan juga *Firebase Authentication*, yang mendukung fitur autentikasi pengguna.



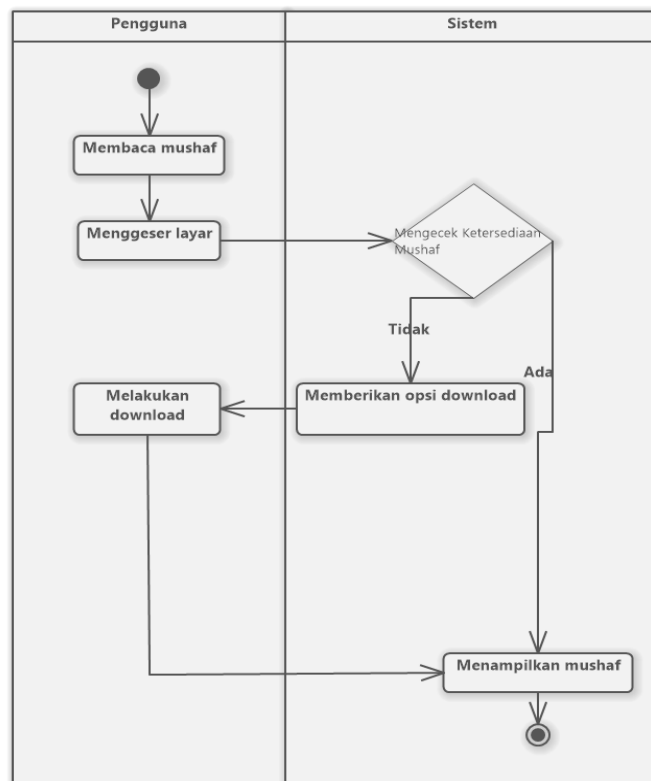
Gambar 3. 11 Gambar relasi database

Gambar 3.11 merupakan relasi *database*, terdapat beberapa komponen di dalam tabel *user* seperti *user_id*, *email*, *password*, dan *session*. *User_id* digunakan untuk memastikan bahwa setiap pengguna yang masuk merupakan pengguna yang unik berdasarkan *user_id* masing-masing. *Email* dan *password* merupakan *email* dan pengguna yang telah mereka gunakan untuk mendaftar. *Session* merupakan tempat penyimpanan sesi milik pengguna yang memiliki korelasi dengan *token* pada tabel *Session* untuk dapat mengakses sesi pengguna.

Pada tabel *Session*, terdapat beberapa komponen di dalamnya, seperti *session_id* yang merupakan identitas atau pembeda setiap sesi. Kemudian ada *login_time* yang merupakan waktu ketika pengguna melakukan *login*. *Expiry_time* yang merupakan waktu kadaluarsa sebuah sesi, dan yang terakhir adalah *token*. Sebagaimana yang sudah dijelaskan pada paragraf sebelumnya *token* merupakan sebuah kunci untuk dapat mengakses sesi pengguna.

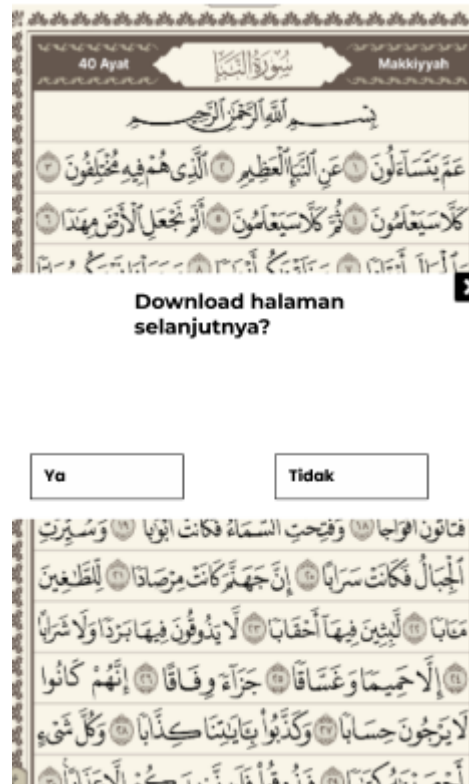
3.2.6 Desain Download

Setelah *usecase* didapatkan mulai dilakukan pembuatan activity diagram untuk *download*,



Gambar 3. 12 Gambar activity diagram download

Gambar 3.12 arsitektur *activity diagram download*. Fitur download baru akan terpicu ketika pengguna membaca mushaf dan menggeser layar aplikasi. Jika pengguna sudah memiliki halaman mushaf, maka pengguna dapat melanjutkan untuk membaca. Jika pengguna belum memiliki mushaf maka pengguna akan diberikan opsi untuk mengunduh mushaf , yang kemudian akan ditampilkan.



Gambar 3. 13 Gambar prototype pop-up download

Gambar 3.13 merupakan gambar desain prototipe *pop-up download*. *Pop-up download* ini sendiri akan muncul ketika pengguna telah membaca halaman mushaf pada juz 30 ataupun juz lainnya selagi tidak terdapat halaman mushaf pada aplikasi. Pengguna dapat menekan ya untuk mengunduh dan tidak jika tidak menginginkan.

3.3 Implementasi

Tahap ini merupakan tahapan implementasi kelanjutan dari tahap sebelumnya (desain). Pada tahap ini terbagi menjadi 4 bagian pertama adalah implementasi *register*, kedua implementasi *login*, ketiga implementasi *logout*, dan keempat implementasi *download*.

Sebelum mengimplementasikan sesi lokal. Pengembang harus memasukkan dependensi *Async Storage* yang memiliki fungsi untuk menyimpan data secara asinkron. Kemudian masukkan dependensi *firebase*, yang berfungsi untuk mengakses fitur-fitur *Firestore*.

```
import AsyncStorage from '@react-native-async-storage/async-storage';
import firebase from 'firebase';
```

Gambar 3. 14 Gambar dependensi penyimpanan

```
const getUserPreference = async () => {  
  setLoading(true)  
  try {  
    const value = await AsyncStorage.getItem('userPreferences');  
    if (value !== null) {  
      dispatch({  
        action: 'SET_ONBOARDING_STATUS',  
        payload: false  
      })  
      userDispatch({  
        action: 'SET_USER_DATA',  
        payload: JSON.parse(value),  
      })  
      return setLoading(false)  
    }  
    dispatch({  
      action: 'SET_ONBOARDING_STATUS',  
      payload: true,  
    })  
    return setLoading(false)  
  } catch (e) {  
    setLoading(false)  
    console.log(e)  
  }  
}
```

Gambar 3. 15 Gambar kode untuk penyimpanan data sesi lokal

Kemudian seperti gambar di atas yang menunjukkan kode untuk menyimpan data secara lokal. *SET_ONBOARDING_STATUS* merupakan sebuah nilai *default* dari sesi lokal,

yang mana sesi masih kosong dan belum diisi data. Jika ada data yang tertulis maka `SET_ONBOARDING_STATUS` akan menjadi `false` dan data sesi lokal akan ditulis.

```
const firebaseConfig = {
  apiKey: "your-api-key",
  authDomain: "your-auth-domain",
  projectId: "your-project-id",
  storageBucket: "your-storage-bucket",
  messagingSenderId: "your-messaging-sender-id",
  appId: "your-app-id"
};

if (!firebase.apps.length) {
  firebase.initializeApp(firebaseConfig);
}
```

Gambar 3. 16 Gambar konfigurasi firebase

Kemudian setelah sesi lokal berhasil terbuat. Pengembang kemudian membuat sebuah dokumen konfigurasi yang disesuaikan dengan kebutuhan seperti Gambar 3.12, untuk dapat mengakses fitur-fitur firebase.

3.3.1 Register

```
const register = async () => {
  try {
    await firebase.auth().createUserWithEmailAndPassword(email, password);
    const user = firebase.auth().currentUser;
    const userToken = user.uid;
    const loginTime = firebase.database.ServerValue.TIMESTAMP;
    const expiryTime = null;
    const hashedPassword = await bcrypt.hash(password, 10);
```

```

const sessionData = {
  session_id: userToken,
  expiry_time: expiryTime,
  login_time: loginTime,
  token: userToken,
};

const userData = {
  user_id: userToken,
  email: email,
  password: hashedPassword,
  session_token: userToken,
};

await firebase.database().ref(`sessions/${userToken}`).set(sessionData);
await firebase.database().ref(`users/${userToken}`).set(userData);
await firebase.database().ref(`users/${userToken}`).update({
session_token: userToken });

await AsyncStorage.setItem('session_token', userToken);
} catch (error) {
  console.error(error);
}
};

```

Gambar 3. 17 Gambar kode register dan penyimpana data akun di database

Gambar 3.13 merupakan kode ketika pengguna melakukan *register*. Kode tersebut berisikan hal-hal yang dibutuhkan ketika *register* dalam hal ini *email* dan *password*. Kemudian, ketika pengguna berhasil untuk melakukan *register* data sesi dan data pengguna terbuat yang terlihat pada *set(sessionData);* dan *set(userData);*, kemudian sesi token akan disimpan pada penyimpanan lokal, jika dibutuhkan sewaktu-waktu menggunakan *await AsyncStorage.setItem('session_token', userToken);*.

3.3.2 Login

Setelah sesi lokal berhasil terbentuk. Peneliti kemudian membuat kode *login* yang memiliki tujuan untuk mengambil data sesi

```
const login = async (email, password) => {  
  try {  
    const [sessionData, setSessionData] = useState(null);  
    await firebase.auth().signInWithEmailAndPassword(email, password);  
    const user = firebase.auth().currentUser;  
    const userToken = user.uid;  
  
    await AsyncStorage.setItem('session_token', userToken);  
    if (sessionData.expiryTime !== null) {  
      Alert.alert('sesi telah berakhir')  
    } else {  
      updateSession();  
    }  
  } catch (error) {  
    console.error(error);  
  }  
};
```

Gambar 3. 18 Gambar kode login

Gambar 3.14 merupakan kode ketika pengguna melakukan *login*. Kode tersebut berisikan kebutuhan yang diambil ketika *login* yakni *email* dan *password*. Setelah pengguna memasukkan *email* dan *password* sesi akan disimpan pada penyimpanan lokal seperti pada fitur sebelumnya. Kemudian, ketika pengguna berhasil untuk *login* data sesi diambil kembali dari database jika sesi belum kadaluarsa yang diindikasikan oleh *if (sessionData.expiryTime*

`!= null`) karena pada dasarnya pengaturan `expiryTime` adalah `null`, dan jika terisi maka sesi telah berakhir.

```
const updateSession = async () => {
  try {
    const snapshot = await sessionRef.once('value');
    const databaseSessionData = snapshot.val();
    await AsyncStorage.setItem('userPreferences',
JSON.stringify(databaseSessionData));

    setSessionData(databaseSessionData);
  } catch (error) {
    console.error(error);
  };
  updateSession();
}

useEffect(() => {
  login()
}, [])
```

Gambar 3. 19 Gambar kode untuk update sesi

Kode pada Gambar 3.15 adalah kode untuk melakukan *update* sesi yang mana kode `AsyncStorage.setItem('userPreferences',JSON.stringify(databaseSessionData));` akan menimpa sesi yang berada pada penyimpanan lokal, meskipun secara *default* jika pengguna *logout* maka sesi lokal akan terhapus. Kemudian sistem akan mengeksekusi kode *login* menggunakan `useEffect(() => {login() },[])`.

```
const getUserPreference = async () => {
  setLoading(true);
  try {
    const value = await AsyncStorage.getItem('userPreferences');
```



```

if (value !== null) {
  dispatch({
    action: 'SET_ONBOARDING_STATUS',
    payload: false,
  });
  userDispatch({
    action: 'SET_USER_DATA',
    payload: JSON.parse(value),
  })
  const userId = firebase.auth().currentUser.uid;
  await firebase.database().ref('session/' + ${userToken}
.set(JSON.parse(value));

return setLoading(false);

```

Gambar 3. 20 Gambar kode simpan sesi ke database

Gambar 3.16 merupakan kode untuk menyimpan sesi lokal ke *database*. Sesi akan disimpan berdasarkan oleh *userId* dari masing-masing pengguna.

3.3.3 Logout

```

const logout = async () => {
  try {
    await firebase.auth().signOut();
    const user = firebase.auth().currentUser;
    const userToken = user.uid;
    const { _, dispatch } = useOnBoardingState()
    const expiryDate = firebase.database.ServerValue.TIMESTAMP;

    await firebase.database().ref(`sessions/${userToken}`).update({
expiry_date: expiryDate });

    if (expiryDate !== null){

```

```

    dispatch({
      action: 'SET_ONBOARDING_STATUS',
      payload: true,
    })
  }
} catch (error) {
  console.error(error);
}
};

useEffect(() => {
  logout()
}, [])

```

Gambar 3. 21 Gambar kode logout

Ketika pengguna melakukan *logout* sebagaimana seperti gambar 3.17, dalam kode ini, *expiryDate* berfungsi untuk menentukan waktu kadaluarsa dari sesi ketika pengguna sudah melakukan *logout*. Kemudian ketika *expiryDate* yang normalnya adalah *null* terisi maka *payload* untuk *SET_ONBOARDING_STATUS* menjadi *true* dan menyebabkan data sesi lokal menjadi *default* atau kembali menjadi kosong.

3.3.4 Download

Setelah melakukan pengembangan *logout*. Diteruskan kepada implementasi *download*.

```

import * as FileSystem from 'expo-file-system';
import * as firebase from 'firebase';

const allowedJuzNumbers = [29, 30];

const juzFileRanges = {
  30: { start: 581, end: 603 },
  29: { start: 560, end: 580 },
};

```

```
let lastDownloadedJuzNumber = 30;
let lastDownloadedFileNumber = 581;

const baseFolderPath =
`${FileSystem.documentDirectory}assets/mushaf/juz/pages`;

const downloadData = async () => {
  try {
    const juzNumber = lastDownloadedJuzNumber - 1;

    if (allowedJuzNumbers.includes(juzNumber)) {
      const juzRange = juzFileRanges[juzNumber];
      const fileNumber = lastDownloadedFileNumber - 1;

      const juzFolderName = `juz${juzNumber}`;
      const folderPath = `${baseFolderPath}/${juzFolderName}`;

      if (fileNumber >= juzRange.start && fileNumber <= juzRange.end) {
        const folderExists = await FileSystem.getInfoAsync(folderPath);

        if (folderExists.exists) {
          const imageFileName = `page_${fileNumber}.jpg`;
          const imagePath = `${folderPath}/${imageFileName}`;
          const fileExists = await FileSystem.getInfoAsync(imagePath);

          if (fileExists.exists) {
            lastDownloadedFileNumber--;

            if (fileNumber === juzRange.end) {
```

```
        lastDownloadedJuzNumber--;  
    }  
    } else {  
        const imageUrl = await  
firebase.storage().ref(`mushaf/${imageFileName}`).getDownloadURL();  
  
        const response = await FileSystem.downloadAsync(  
            imageUrl,  
            imagePath  
        );  
  
        if (response.status === 200) {  
  
            lastDownloadedFileNumber--;  
  
            if (fileNumber === juzRange.end) {  
                lastDownloadedJuzNumber--;  
            }  
        } else {  
            Alert.alert('Error', 'Terjadi kesalahan saat mengunduh  
gambar.');
```

```

    } catch (error) {
        console.error('Error mengunduh gambar:', error);
        Alert.alert('Error', 'Terjadi kesalahan saat mengunduh gambar.');
```

Gambar 3. 22 Gambar kode unduh mushaf

Gambar 3.18 menunjukkan kode untuk mengunduh data mushaf dari *database*. setelah melakukan *import* perlu dideklarasikan nomer juz yang diperbolehkan dan juga jarak masing-masing juz dengan menambahkan *allowedJuzNumbers* dan *juzFileRange*. Secara *default* aplikasi sudah memiliki mushaf di juz 30, sehingga *lastDownloadedJuzNumber* dan *lastDownloadedFileNumber* disesuaikan dengan juz 30 dan juga halaman pertama pada juz 30. Kemudian sistem akan melakukan pengecekan terhadap nomor juz apakah masuk kedalam *range* atau tidak dengan (*allowedJuzNumbers.includes(juzNumber)*). Kemudian melakukan pengecekan nomor *file* mushaf masuk ke dalam *range* angka halaman (*fileNumber >= juzRange.start &&fileNumber <= juzRange.end*). Jika masuk, maka sistem akan melakukan pengecekan *folder* juz pada penyimpanan lokal dan jika ada akan diteruskan ke *folder file* dari mushaf. Jika *file* tidak terdapat pada folder maka sistem akan mengunduh mushaf dan mengupdate *lastDownloadedFileNumber*.

3.4 Pengujian

Aplikasi akan melewati pengujian, berdasarkan dari tahapan sebelumnya (implementasi). Pengujian akan dilakukan dengan menggunakan *blackbox testing* dan juga *user acceptance test*.

3.4.1 *Blackbox Testing*

Pada tahap ini akan dilakukan pengujian dengan menggunakan *blackbox testing* untuk menguji fungsionalitas. Pengujian akan dilakukan dengan menguji fitur-fitur atau fungsionalitas dari aplikasi. Pengujian untuk masing-masing fitur dan fungsionalitas akan dilakukan sebanyak 7 kali. Hasil pengujian akan dibuatkan tabel yang berisikan skenario pengujian, kasus pengujian, hasil yang diharapkan, hasil pengujian, dan juga keterangan.

3.4.2 *User Acceptance Test(UAT)*

Pengujian ini akan dilakukan kepada 6 responden. Keenam responden tersebut setidaknya pernah menghafal Al-Qur'an sebanyak 1 juz, dan dapat mengoperasikan gawai. pengujian akan dilakukan secara *remote* dengan menggunakan *software Team Viewers* yang memiliki fitur *remote control* agar pengguna lain dapat melakukan kontrol emulator android secara *remote*, dari tempat masing-masing. Pengujian juga akan dilakukan dengan menggunakan *software Discord*, dengan memanfaatkan fitur *share screen* pada *software* tersebut. Kemudian setelah responden selesai melakukan pengujian, responden akan diberikan kuesioner menggunakan *google form*. Kuesioner tersebut berisikan pertanyaan-pertanyaan berdasarkan kebutuhan yang telah dirancang dan juga ekspektasi pengguna terhadap aplikasi. Dari kuesioner tersebut kemudian akan ditentukan *acceptance criteria coverage* berdasarkan hasil bobot yang telah didapat pada kuesioner.

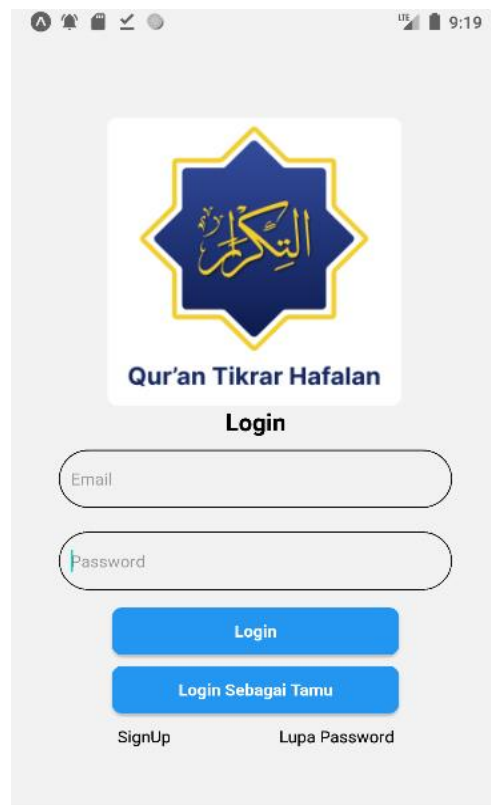
BAB IV

HASIL DAN PEMBAHASAN

4.1 Fitur

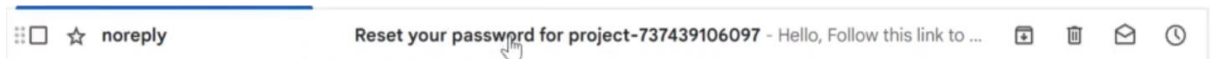
Dalam pengembangan aplikasi yang telah dilaksanakan terdapat empat fitur penunjang personalisasi proses menghafal untuk memudahkan proses menghafal yang telah peneliti tambahkan pada aplikasi. Empat fitur tersebut terdiri dari fitur *login*, fitur *register*, fitur *logout*, dan fitur *download*.

4.1.1 Fitur Login



Gambar 4. 1 Gambar Antarmuka login

Gambar 4.1 merupakan Antarmuka dari halaman *login*. Pada fitur *login* pengguna dapat untuk memasukkan data berupa *email* dan *password* untuk membuat akun. Kemudian pengguna dapat menekan tombol masuk sebagai tamu, namun datanya tidak tersimpan pada *database*.



Gambar 4. 2 Gambar email untuk reset password

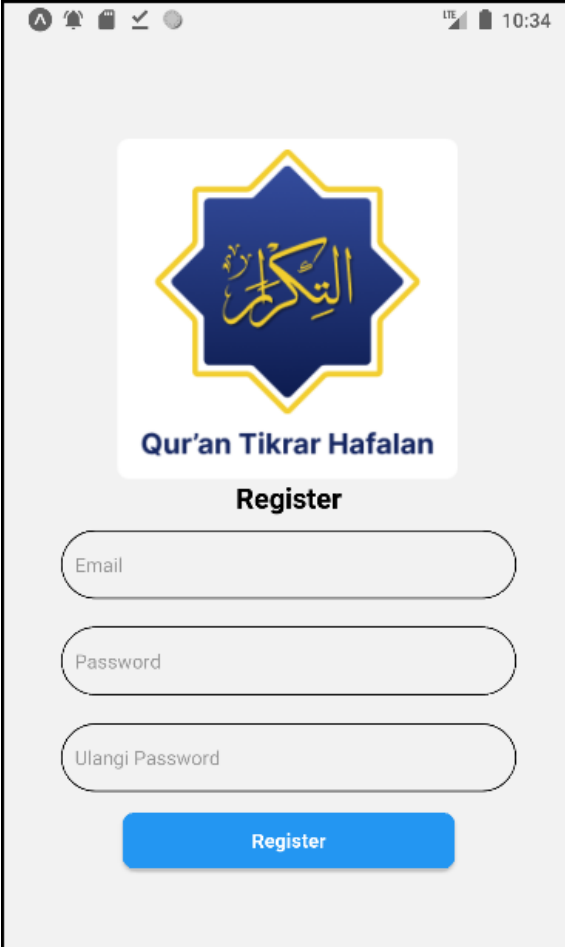
Pengguna juga dapat untuk menekan lupa *password* jika pengguna lupa akan kata sandi miliknya. Pengguna akan mendapatkan email seperti pada Gambar 4.2.



Gambar 4. 3 Gambar antarmuka homepage

Akun diperlukan untuk mengakses penyimpanan di *database*. Gambar 4.2 merupakan halaman utama yang berisikan perkembangan hafalan pengguna, setelah melakukan login.

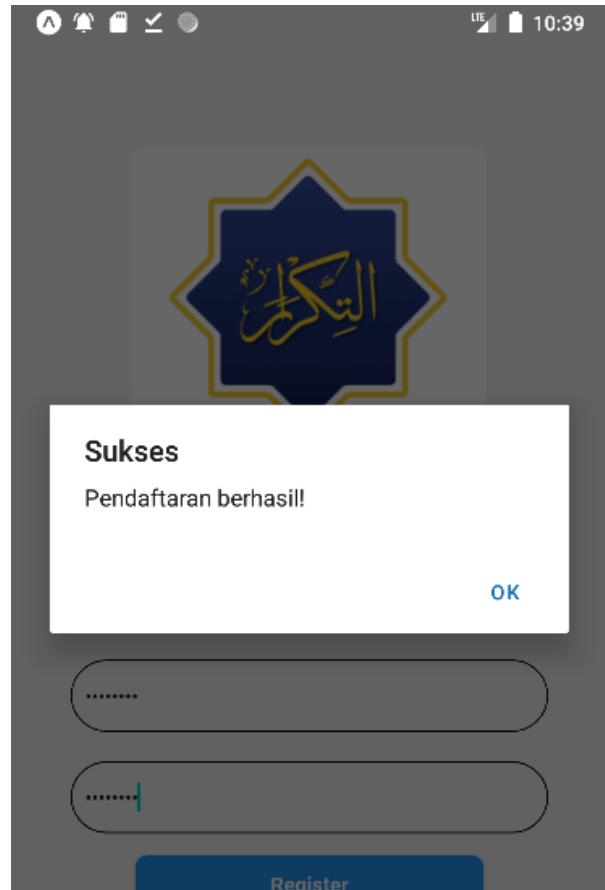
4.1.2 Fitur Register



The image shows a mobile application interface for registration. At the top, there is a status bar with icons for signal, battery, and time (10:34). Below the status bar is a logo consisting of a blue star with yellow Arabic calligraphy inside, and the text "Qur'an TIKRAR Hafalan" below it. Underneath the logo is the word "Register" in bold. There are three input fields: "Email", "Password", and "Ulangi Password". At the bottom, there is a blue button labeled "Register".

Gambar 4. 4 Gambar antarmuka register

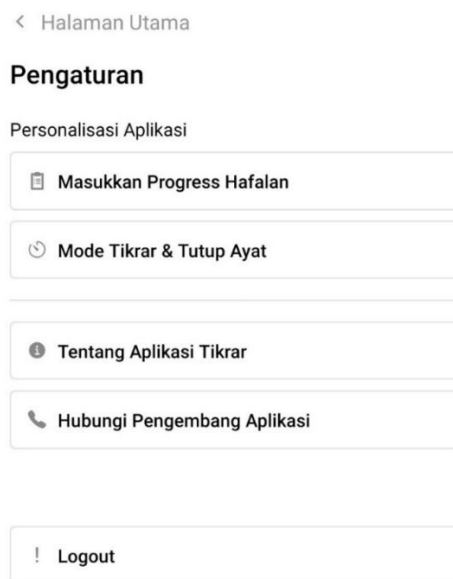
Gambar 4.4 merupakan antarmuka dari halaman *register*. Halaman *register* sendiri dapat diakses dengan menekan tulisan “*SignUp*” pada halaman *login*. Pada halaman ini pengguna dapat menuliskan *email* serta *password* mereka untuk mendaftarkan akun, kemudian pengguna harus mengulangi *password* dengan tepat untuk dapat mendaftarkan akun.



Gambar 4. 5 Gambar notifikasi ketika pendaftaran sukses

Gambar 4.5 menunjukkan notifikasi yang akan pengguna dapat ketika pengguna telah berhasil untuk mendaftar.

4.1.3 Fitur Logout



Gambar 4. 6 Gambar halaman pengaturan ketika pengguna sudah login

Gambar 4.6 menunjukkan fitur *logout* yang terdapat pada halaman pengaturan. Fitur *logout* sendiri memiliki fungsi untuk mengeluarkan pengguna yang sudah *login*. Fitur ini juga memiliki fungsi untuk menghapus sesi yang ada pada perangkat pengguna.

4.1.4 Fitur Download



Gambar 4. 7 Gambar notifikasi download

Pada gambar 4.9 terlihat sebuah notifikasi yang muncul. Notifikasi ini akan muncul Ketika pengguna telah kehabisan mushaf pada perangkat mereka. Sehingga sistem menawarkan kepada pengguna untuk mendownload lembaran mushaf selanjutnya.

4.2 Pengujian

Pengujian dilaksanakan untuk mengetahui apakah aplikasi yang dikembangkan oleh peneliti sudah sesuai dengan apa yang direncanakan . Pengujian yang telah dilaksanakan adalah pengujian fungsionalitas dan *user acceptance test* (UAT).

4.2.1 Blackbox Testing

Pada tahap ini telah dilakuan *blackbox testing* dengan menguji fungsionalitas. Pengujian ini dilakukan terhadap fitur-fitur aplikasi dapat dilihat pada Tabel 4.1.

Tabel 4. 1 Tabel hasil uji fungsionalitas

NO.	<i>Skenario Pengujian</i>	<i>Kasus Pengujian</i>	<i>Hasil yang diharapkan</i>	<i>Hasil pengujian</i>	<i>Keterangan</i>
1.	pengujian <i>signup</i> pengguna berhasil melakukan registrasi ke sistem	Melakukan registrasi dan, sistem mengarahkan pengguna supaya bisa mengakses halaman utama	Pengguna dapat untuk melakukan registrasi dan masuk ke halaman utama	Berhasil 7/7 Kali	
2.	Melakukan <i>Login</i> dan data berhasil dikembalikan	Pengguna melakukan <i>login</i> dan sistem kemudian mengarahkan pengguna ke halaman utama dengan halaman utama yang sudah sesuai dengan data sesi di <i>database</i>	Pengguna berhasil untuk melakukan <i>login</i> dan masuk ke halaman utama dengan data yang telah dikembalikan	Berhasil 6/7 kali	Terjadi kegagalan karena adanya data yang tidak terambil
3.	Pengujian <i>logout untuk keluar</i>	Pengguna keluar dari akun dan sesi lokal berhasil terhapus	Pengguna berhasil keluar dan data sesi pada aplikasi pengguna berhasil dihapus	Berhasil 6/7 kali	Terjadi kegagalan karena data sesi tidak terhapus sepenuhnya.

NO.	<i>Skenario Pengujian</i>	<i>Kasus Pengujian</i>	<i>Hasil yang diharapkan</i>	<i>Hasil pengujian</i>	<i>Keterangan</i>
4.	Pengujian <i>download</i> Untuk mendownload mushaf	Pengguna menekan tombol “ya” pada notifikasi <i>download</i> untuk mengunduh mushaf yang belum ada di perangkat mereka	Pengguna berhasil mengunduh mushaf yang belum terunduh di perangkatnya	Berhasil 7/7 kali	

Berdasarkan Tabel 4.1 fitur-fitur aplikasi dapat berjalan sesuai dengan rencana dengan total tingkat keberhasilan 26 kali dari total 28 kali percobaan. Tingkat presentase keberhasilan dari pengujian ini adalah sebesar 92,8%. Sehingga dapat disimpulkan bahwa aplikasi dapat berjalan sesuai dengan rencana.

4.2.2 *User Acceptance Test (UAT)*

Setelah sesi percobaan selesai, pengguna diberikan sebuah kuesioner berupa pertanyaan pada tabel 4.2. Kuesioner tersebut memiliki bobot yang dapat dilihat pada tabel 4.3.

Tabel 4. 2 Tabel pertanyaan kuesioner

No.	Pertanyaan
1	Apakah antarmuka pengguna (UI) mudah dipahami dan ramah pengguna?
2	Apakah pengguna merasa bahwa data hafalan mereka dapat dikelola dengan baik?
3	Apakah fitur-fitur yang ditawarkan sesuai dengan harapan pengguna?
4	Apakah informasi yang ditampilkan dalam aplikasi sesuai dengan yang diharapkan oleh pengguna?

Tabel 4. 3 Tabel bobot penilaian kuesioner

No	Keterangan	Bobot
1	Sangat tidak setuju	1
2	Tidak setuju	2
3	Ragu	3
4	Setuju	4
5	Sangat setuju	5

Berdasarkan pengujian yang telah dilakukan kepada 6 orang responden maka didapati hasil seperti pada tabel 4.4.

Tabel 4. 4 Tabel hasil penilaian kuesioner

No. Pertanyaan	Penilaian					Bobot penilaian/bobot maksimum
	1	2	3	4	5	
1	-	-	-	4x5=20	5	25/30
2	-	-	-	4x2=8	5x4=20	28/30
3	-	-	-	4x4=16	5x2=10	26/30
4	-	-	-	4x2=8	5x4= 20	28/30

Dari tabel di atas terlihat adanya tingkat bobot sebesar 106/120. Kemudian dari bobot tersebut kemudian dicarilah *Acceptance Criteria Coverage* dengan menggunakan rumus ACC. Berdasarkan rumus tersebut maka peneliti mendapatkan jumlah nilai ACC atau kriteria

kepuasan pengguna sebesar $106/120 \times 100\% = 88,3\%$. Dari hasil presentase tersebut sudah mendekati angka 100%, yang menunjukkan bahwa aplikasi sudah dapat diterima oleh pengguna.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Dari penelitian Implementasi Personalisasi dengan Menggunakan Sesi untuk Menyimpan Data Hafalan Pengguna dalam Aplikasi Al-Qur'an Berbasis *Mobile*, dapat disimpulkan sebagai berikut:

- a. Masalah yang akan dijawab dari penelitian ini adalah bagaimana cara mengimplementasikan personalisasi dalam sebuah aplikasi Al-Qur'an berbasis *mobile* dengan menggunakan sesi untuk menyimpan data hafalan pengguna, dengan tujuan membuat personalisasi progress menghafal melalui aplikasi Al-Qur'an berbasis *mobile* untuk memudahkan proses menghafal. Berdasarkan hasil pengujian dan implementasi menunjukkan bahwa aplikasi telah mampu memberikan fitur personalisasi untuk memudahkan proses menghafal pengguna.
- b. Masalah yang akan dijawab dari penelitian ini adalah Apakah personalisasi dapat membantu memudahkan proses menghafal?, dengan tujuan membuat personalisasi progress menghafal melalui aplikasi Al-Qur'an berbasis *mobile* untuk memudahkan proses menghafal. Berdasarkan hasil pengujian dan implementasi didapati bahwa aplikasi dapat membantu memudahkan proses menghafal jika dilihat dari tingkat presentase *Acceptance Criteria Coverage (ACC)* sebesar 88,3% dan kesuksesan fitur sebesar 92,8%.

5.2 Saran

Berdasarkan penelitian yang telah peneliti lakukan, maka peneliti memiliki beberapa saran yang telah peneliti dapatkan dari hasil penelitian dan pengujian, berikut adalah saran peneliti.

- a. Menggunakan database dengan Bandwidth yang tinggi supaya mempercepat proses download bagi pengguna.
- b. Memberikan opsi download yang lebih banyak, disesuaikan dengan kemampuan database.

- c. Menambahkan fitur-fitur penunjang keamanan akun dan data karena untuk sekarang masih begitu rawan.

DAFTAR PUSTAKA

- Adalyn, N. (2022, April 19). Retrieved from LinkedIn: <https://www.linkedin.com/pulse/advantages-disadvantages-firebase-nav-adalyn>
- Alamdari, N., Yaraganalu, S., & K. N. (2018). A Real-Time Personalized Noise Reduction Smartphone App for Hearing Enhancement. *IEEE*. doi: <https://doi.org/10.1109/spmb.2018.8615620>
- Andriana, M. (2020, October 28). Retrieved from Binus: <https://sis.binus.ac.id/2020/10/28/user-acceptance-test/>
- Ardito, L., Coppola, R., Malnati, G., & Torchiano, M. (2020, November). Effectiveness of Kotlin vs. Java in android app development tasks. doi: [dx.doi.org/10.1016/j.infsof.2020.106374](https://doi.org/10.1016/j.infsof.2020.106374)
- AWS. (n.d.). Retrieved from aws.amazon: <https://aws.amazon.com/what-is/sdlc/>
- Aziz, M., Abdullah, W., Ahmad, A. M., Amat Mushim, M., & Shahrudin, M. S. (2019). Comparison between Conventional Method and Modern Technology in Al-Qur'an Memorization.
- Bose, S. (2018, June 20). A COMPARATIVE STUDY: JAVA VS KOTLIN PROGRAMMING IN ANDROID APPLICATION DEVELOPMENT. *International Journal of Advanced Research in Computer Science. IJARCS*. doi: <http://dx.doi.org/10.26483/ijarcs.v9i3.5978>
- Danielsson, W. (2016). React Native application development. *Linköpings universitet*.
- Dynatrace. (2024, January 2). Retrieved from Dynatrace: <https://docs.dynatrace.com/docs/platform-modules/digital-experience/rum-concepts/user-session>
- Eksan, M. (2023, April 9). Retrieved from Kuasarakyat: <https://kuasarakyat.com/hafidz-al-quran-indonesia-menang-dari-arab-saudi-kalah-dari-mesir/>
- Eshetu Fentaw, A. (2020). Cross platform mobile application development: a comparison study of React Native Vs Flutter. *UNIVERSITY OF JYVÄSKYLÄ FACULTY OF INFORMATION TECHNOLOGY*.
- Everett, G. D., & McLeod, R. (2007). *Software Testing: Testing Across the Entire Software Development Life Cycle*. Wiley-IEEE Computer Society Pr EBooks. Retrieved from <http://ci.nii.ac.jp/ncid/BA89619214>
- Fatah. (2023, March 29). Retrieved from Rumahweb: <https://www.rumahweb.com/journal/session-adalah/>

- Geeksforgeeks. (2023, December 6). Retrieved from geeksforgeeks: <https://www.geeksforgeeks.org/software-testing-functional-testing/?ref=lbp>
- H, O. (2023, october 5). Retrieved from LinkedIn: <https://www.linkedin.com/pulse/comprehensive-guide-software-development-life-cycle-sdlc-obert-h->
- Hanna, K. T., & Rosencrance, L. (2023, May 19). Retrieved from Techtarget: <https://www.techtarget.com/searchmobilecomputing/definition/Google-Firebase>
- Haviluddin, H. (2016). Memahami penggunaan UML (Unified Modelling Language). *Informatika Mulawarman : Jurnal Ilmiah Ilmu Komputer*, 1–15. doi:<https://doi.org/10.30872/jim.v6i1.16>
- IBM. (2021, 03 06). Retrieved from IBM: <https://www.ibm.com/docs/en/sva/9.0.3?topic=servers-user-session-management-concepts>
- Ifana, S. R. (2023). The Influence of Al-Qur'an Memory Level and Learning Motivation on Student Academic Achievement. *JIEE*.
- Islam, D. M., & Mazumder, T. (2010). Mobile application and its global impact. *International Journal of Engineering & Technology*, 72-78.
- Khurram Khan, M., & M. Alginahi, Y. (2013). The Holy Quran Digitization: Challenges and Concerns. *Life Science Journal*.
- Kucirkova, N. (2019). Children's agency by design: Design parameters for personalization in story-making apps. *International Journal of Child-Computer Interaction*, 21, 112–120. doi:<https://doi.org/10.1016/j.ijcci.2019.06.003>
- Mehrzadi, D., & Feitelson, D. G. (2012). On extracting session data from activity logs. *5th Annual International Systems and Storage Conference SYSTOR*. doi:<https://doi.org/10.1145/2367589.2367592>
- Meilinaeka, M. (2023, July 13). Retrieved from Telkom University: <https://it.telkomuniversity.ac.id/perbedaan-uat-dan-sit/#:~:text=Pengujian%20UAT%20sendiri%20yaitu%20berupa,akhir%20dari%20proses%20software%20testing.>
- MongoDB. (n.d.). Retrieved from MongoDB: [https://www.mongodb.com/nosql-explained#:~:text=NoSQL%20databases%20\(aka%20%22not%20only,wide%2Dcolumn%2C%20and%20graph.](https://www.mongodb.com/nosql-explained#:~:text=NoSQL%20databases%20(aka%20%22not%20only,wide%2Dcolumn%2C%20and%20graph.)

- Nielsen Norman Group. (n.d.). Retrieved from Nielsen Norman Group: <https://www.nngroup.com/articles/customization-personalization/>
- Optimizely. (n.d.). Retrieved from Optimizely: <https://www.optimizely.com/optimization-glossary/personalization/>
- Prechelt, L. (2000). An empirical comparison of seven programming languages. *IEEE*. doi:<http://dx.doi.org/10.1109/2.876288>
- Raghib, A.-S. (2007). Cara Cerdas Hafal AlQur'an. *Solo : Aqwam*, hal. 53.
- Rahmalia, N. (2021, February 14). Retrieved from Glints: <https://glints.com/id/lowongan/black-box-testing/>
- Rice, R. (n.d.). Retrieved from Riceconsulting: <https://www.riceconsulting.com/home/index.php/User-Acceptance-Testing/what-is-the-value-of-user-acceptance-testing.htm>
- Setiawan, R. (2021, November 1). Retrieved from Dicoding: <https://www.dicoding.com/blog/black-box-testing/>
- Smutny, P. .. (2012). Mobile development tools and cross-platform solutions. *International Carpathian Control Conference (ICCC), IEEE*. IEEE.
- Susanto, Y., Djojosoewarno, P., & Rosnaeni, R. (2009). *Pengaruh olahraga ringan terhadap memori jangka pendek pada wanita dewasa*. Maranatha Journal of Medicine and Health. Retrieved from <https://www.neliti.com/publications/150079/pengaruh-olahraga-ringan-terhadap-memori-jangka-pendek-pada-wanita-dewasa>
- Tan, C. W., Ling, L., Yu, P., Hang, C. N., & Wong, M. (2020). Mathematics Gamification in Mobile App Software. *IEEE*.
- Tomlin, C. (2023, November 17). Retrieved from Cedars-sinai: <https://www.cedars-sinai.org/blog/concerned-about-memory-loss.html#:~:text=Smoking%2C%20exposure%20to%20air%20pollution,is%20really%20an%20attention%20problem>
- Trivedi, J., & Trivedi, H. (2018). Investigating the Factors That Make a Fashion App Successful: The Moderating Role of Personalization. *Journal of Internet Commerce*, 170–187. doi:<https://doi.org/10.1080/15332861.2018.1433908>
- Wu, W. (2018). React Native vs Flutter, cross-platform mobile application frameworks. *Metropolia*.
- Yuwanto, E. (2010). Retrieved from Republika Online: <https://khazanah.republika.co.id>

Zakariah, M., Khan, M. K., Tayan, O., & Salah, K. (2017). Digital Quran Computing: Review, Classification, and Trend Analysis. *COMPUTER ENGINEERING AND COMPUTER SCIENCE*, 3077–3102.

LAMPIRAN