

**PENGEMBANGAN APLIKASI AL-QUR'AN DIGITAL
BERBASIS WEB GUNA MEMFASILITASI PEMBACA DAN
PENGHAFAL AL-QUR'AN**

(Skripsi)

Oleh:

**ALDO WIJAYA
2015061012**



**FAKULTAS TEKNIK
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2024**

**PENGEMBANGAN APLIKASI AL-QUR'AN DIGITAL
BERBASIS WEB GUNA MEMFASILITASI PEMBACA DAN
PENGHAFAL AL-QUR'AN**

Oleh

ALDO WIJAYA

Skripsi

**Sebagai Salah Satu Syarat Untuk Mencapai Gelar
SARJANA TEKNIK**

Pada

**Program Studi Teknik Informatika
Fakultas Teknik Universitas Lampung**



**FAKULTAS TEKNIK
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2024**

ABSTRAK

PENGEMBANGAN APLIKASI AL-QUR'AN DIGITAL BERBASIS WEB GUNA MEMFASILITASI PEMBACA DAN PENGHAFAL AL-QUR'AN

Oleh

ALDO WIJAYA

Pengaruh perkembangan teknologi informasi tidak hanya terbatas dalam hal pekerjaan, melainkan juga dalam hal keagamaan. Terdapat banyak sekali pemanfaatan teknologi informasi yang telah diimplementasikan untuk mempermudah aktivitas keagamaan masyarakat di dunia, salah satunya adalah Al-Qur'an digital. Al-Qur'an digital merupakan salah satu bentuk pemanfaatan teknologi yang dapat meningkatkan kualitas pemahaman dan pembelajaran kitab suci Al-Qur'an umat islam di dunia. Hadirnya Al-Qur'an digital memberikan kemudahan dari segi ruang dan waktu bagi umat islam yang hendak mempelajari dan mengamalkan perintah yang ada di dalam Al-Qur'an. Penelitian ini bertujuan untuk mengembangkan aplikasi Al-Qur'an digital berbasis web yang dapat memfasilitasi pembaca dan penghafal Al-Qur'an. Kemudian aplikasi yang telah dikembangkan juga diuji dari sisi fungsionalitas dan nilai *usability* atau kebergunaan. Adapun metode pengembangan sistem pada penelitian ini adalah *personal extreme programming* (PXP). Proses pengujian aplikasi ini menggunakan metode *black box testing*, *system usability scale*, dan *heuristic evaluation*. Hasil penelitian ini menunjukkan bahwa aplikasi Al-Qur'an digital telah berhasil dikembangkan dengan menggunakan metode *personal extreme programming* dengan durasi 7 iterasi atau 53 hari. Pengujian menggunakan metode *system usability scale* pada penelitian ini melibatkan 13 orang dengan nilai akhir 88,46 dan pengujian menggunakan metode *heuristic evaluation* pada penelitian ini melibatkan 3 orang dengan skor akhir *severity rating* 0,03. Kedua pengujian tersebut telah memenuhi syarat aplikasi dapat dikatakan *acceptable*. Produk akhir penelitian ini dapat diakses melalui *link qurtal.netlify.app*.

Kata kunci: Al-Qur'an, digital, *Personal Extreme Programming*

ABSTRACT

DEVELOPMENT OF A WEB-BASED DIGITAL AL-QUR'AN APPLICATION TO FACILITATE READING AND MEMORIZING THE AL-QUR'AN

By

ALDO WIJAYA

The influence of developments in information technology is not only limited to work, but also to religious matters. There are many uses of information technology that have been implemented to facilitate the religious activities of people in the world, one of which is the digital Al-Qur'an. Digital Al-Qur'an is a form of technology utilization that can improve the quality of understanding and learning of the holy book of Al-Qur'an for Muslims in the world. The presence of the digital Al-Qur'an provides convenience in terms of space and time for Muslims who want to study and practice the commands in the Al-Qur'an. This research aims to develop a web-based digital Al-Qur'an application that can facilitate readers and memorizers of the Al-Qur'an. Then the application that has been developed is also tested in terms of functionality and usability or usefulness. The system development method in this research is personal extreme programming (PXP). The application testing process uses the black box testing method, system usability scale, and heuristic evaluation. The results of this research show that the digital Al-Qur'an application has been successfully developed using the personal extreme programming method with a duration of 7 iterations or 53 days. Testing using the system usability scale method in this study involved 13 people with a final score of 88.46 and testing using the heuristic evaluation method in this study involved 3 people with a final severity rating score of 0.03. Both tests have met the application requirements and can be said to be acceptable. The final product of this research can be accessed via link qurtal.netlify.app.

Keywords: Al-Qur'an, digital, Personal Extreme Programming

Judul Skripsi

: PENGEMBANGAN APLIKASI AL-QUR'AN
DIGITAL BERBASIS WEB GUNA
MEMFASILITASI PEMBACA DAN
PENGHAFAL AL-QUR'AN

Nama Mahasiswa

: Aldo Wijaya

Nomor Pokok Mahasiswa

: 2015061012

Program Studi

: S1 Teknik Informatika

Jurusan

: Teknik Elektro

Fakultas

: Teknik



Pembimbing Utama

Dr. Eng. Mardiana, S.T., M.T., IPM.
NIP. 19721603 199903 2 002

Pembimbing Pendamping

Puput Budi Wintoro, S.Kom, M.T.I.
NIP. 19841031 201903 1 004

2. Mengetahui

Ketua Jurusan Teknik Elektro

Herlinawati, S.T., M.T.
NIP. 19710314 199903 2 001

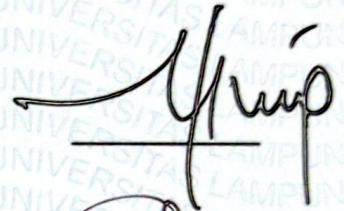
Ketua Program Studi
Teknik Informatika

Yessi Mulyani, S.T., M.T.
NIP. 19731226 200012 2 001

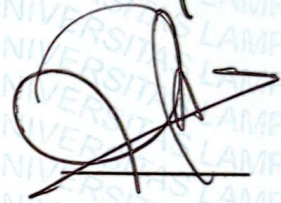
MENGESAHKAN

1. Tim Penguji

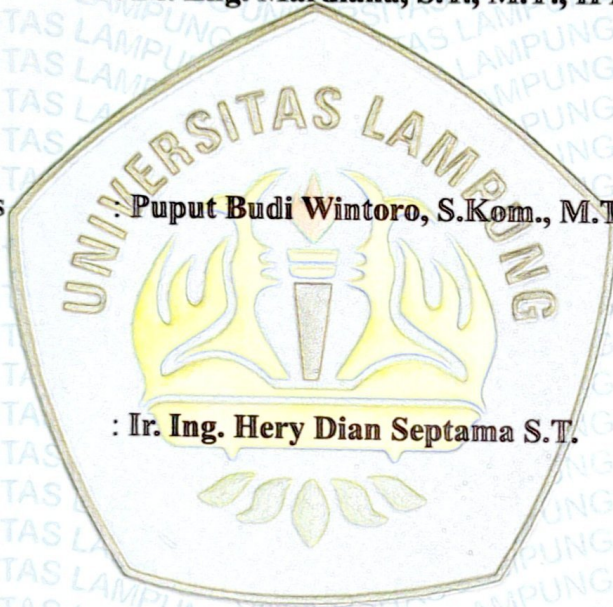
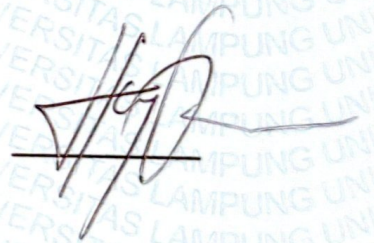
Ketua : Dr. Eng. Mardiana, S.T., M.T., IPM



Sekretaris : Puput Budi Wintoro, S.Kom., M.T.I.



Penguji : Ir. Ing. Hery Dian Septama S.T.



2. Dekan Fakultas Teknik



Dr. Eng. Ir. Helmy Fitriawan, S.T., M.Sc. y

NIP. 19750928 200112 1 002

Tanggal Lulus Ujian Skripsi : 22 Januari 2024

SURAT PERNYATAAN

Saya yang bertanda tangan di bawah ini, menyatakan bahwa skripsi dengan judul “Pengembangan Aplikasi Al-Qur’an Digital Berbasis Web guna Memfasilitasi Pembaca dan Penghafal Al-Qur’an” dibuat oleh saya sendiri. Semua hasil yang terdapat dalam skripsi ini telah mengikuti kaidah penulisan karya tulis ilmiah Universitas Lampung. Apabila pada kemudian hari terbukti bahwa skripsi ini merupakan Salinan atau dibuat oleh orang lain, maka saya bersedia menerima sanksi dengan ketentuan hukum atau akademik yang berlaku.

Bandar Lampung, 22 Januari 2024

Pembuat pernyataan,



Aldo Wijaya

NPM. 2015061012

RIWAYAT HIDUP



Penulis lahir di Bandar Lampung pada tanggal 27 September 2001 dari pasangan Bapak Sodri Alfian, S.E. dan Ibu Umiyati. Penulis telah menyelesaikan pendidikan formal di SD Al-Azhar 2 Bandar Lampung pada tahun 2014, MTsN 2 Bandar Lampung pada tahun 2017, dan SMAN 5 Bandar Lampung pada tahun 2020. Pada tahun 2020, penulis terdaftar sebagai mahasiswa Program Studi Teknik Informatika Universitas Lampung melalui jalur SNPMTN. Selama menjadi mahasiswa, penulis telah aktif dalam berbagai kegiatan dan meraih penghargaan. Kegiatan dan penghargaan tersebut di antaranya adalah:

1. Menjuarai lomba karya tulis ilmiah sebanyak 7 kali (6 tingkat nasional dan 1 tingkat jurusan).
2. Menjuarai lomba pemrograman web sebanyak 2 kali (1 tingkat nasional dan 1 tingkat provinsi).
3. Menjadi pembicara atau pemateri tentang beasiswa, penulisan ilmiah, dan pemrograman sebanyak 6 kali (5 tingkat nasional dan 1 tingkat jurusan).
4. Menjadi mahasiswa berprestasi ke-1 program sarjana Fakultas Teknik, Universitas Lampung pada tahun 2023.
5. Penerima penghargaan mahasiswa berprestasi nasional dari Universitas Lampung pada tahun 2022 dan 2023.
6. Mengikuti program kampus merdeka sebanyak 3 kali (KMMI pada tahun 2021, Studi Independen MSIB pada tahun 2022, dan Magang MSIB pada tahun 2023).
7. Penerima Beasiswa Unggulan (kategori masyarakat berprestasi) pada tahun 2021 - 2024 oleh Kemendikbud Ristek.
8. Penerima penghargaan Duta Inspirasi Indonesia mewakili provinsi Lampung pada tahun 2022 oleh Kemenpora RI.
9. Mengikuti berbagai *internship* yang sejalan dengan fokus karir penulis serta menjadi asisten lab teknik komputer.
10. Turut serta dalam kegiatan organisasi kampus maupun luar kampus.

MOTTO

“Jangan pernah takut bercita-cita untuk menjadi yang terhebat, ingatlah bahwa komputer super hebat saat ini dulunya berasal dari 2 digit angka (0 dan 1)”

- Aldo Wijaya

"Keberhasilan kita tergantung pada sinergi dan kerja sama. Mari bersama-sama, kita bentuk jalan menuju puncak prestasi."

- ChatGPT

PERSEMBAHAN

Segala puji bagi Allah SWT. Tuhan Yang Maha Esa, karena atas nikmat dan karunia-Nya, Saya selaku penulis dapat menyelesaikan skripsi ini tepat pada waktunya.

Kupersembahkan Skripsi ini Kepada:

Diriku Sendiri

“Yang telah berjuang dalam menempuh perjalanan di dunia kampus dan menyelesaikan skripsi ini sesuai dengan waktu dan tujuannya.”

SANWACANA

Puji syukur kehadiran Allah SWT. yang telah memberikan rahmat dan hidayat-Nya kepada penulis, sehingga penulis dapat menyelesaikan skripsi ini tepat waktu. Salawat serta salam penulis sanjungkan kepada Nabi dan Rasul Muhammad SAW yang penulis harapkan syafaatnya di hari akhir kelak.

Skripsi yang berjudul “Pengembangan Aplikasi Al-Qur’an Digital Berbasis Web guna Memfasilitasi Pembaca dan Penghafal Al-Qur’an” ini disusun sebagai salah satu syarat untuk memperoleh gelar sarjana teknik pada Program Studi Teknik Informatika, Universitas Lampung.

Dalam proses penelitian ini, sangat banyak orang-orang yang terlibat dalam pelaksanaannya. Oleh sebab itu, penulis mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Kedua orang tua dan keluarga yang telah menjadi tempat pulang ternyaman bagi penulis selama ini.
2. Bapak Dr. Eng. Helmy Fitriawan, S.T., M.Sc., selaku Dekan Fakultas Teknik Universitas Lampung
3. Ibu Herlinawati, S.T., M.T., selaku Ketua Jurusan Teknik Elektro Universitas Lampung;
4. Ibu Yessi Mulyani, S.T., M.T. selaku Ketua Program Studi Teknik Informatika Universitas Lampung yang telah membantu proses kelancaran pengerjaan penelitian;
5. Ibu Dr. Eng. Ir. Mardiana, S.T., M.T., I.P.M selaku pembimbing utama dan dosen pembimbing akademik yang telah bersedia membimbing penulis selama melaksanakan penelitian ini dan juga melaksanakan proses perkuliahan.
6. Bapak Puput Budi Wintoro, S.Kom., M.T.I selaku pembimbing pendamping yang telah banyak memberikan masukan dalam proses penelitian ini.
7. Bapak Ir. Ing. Hery Dian Septama S.T., selaku penguji yang telah memberikan banyak pelajaran kepada penulis terkait dengan pelaksanaan penelitian ini.

8. Rose yang selalu hadir untuk berkontribusi dan menemani penulis selama proses pelaksanaan penelitian ini.
9. Mahasiswa Jurusan Teknik Elektro angkatan 2019, 2020, 2021, dan 2022 yang telah banyak berkontribusi dan terlibat dalam pelaksanaan penelitian ini.
10. Teman-teman alumni MTsN 2 Bandar Lampung 2017 yang telah bersedia berkontribusi menjadi responden dalam penelitian ini.
11. Seluruh mahasiswa yang terlibat dalam proses penelitian ini yang tidak dapat penulis sebutkan satu persatu namanya.
12. Teman-teman MSIB PT. Len Industri yang telah berbagi ilmu teknis yang tentunya bermanfaat dalam pengembangan sistem pada penelitian ini.
13. ChatGPT, Github Copilot, Github, dan seluruh aplikasi yang membantu mempermudah penulis dalam memikirkan ide dan mencari solusi atas permasalahan yang ditemukan.

Penulis menyadari bahwa penelitian ini masih banyak kekurangannya, baik dari segi penulisan maupun sistem yang dikembangkan. Oleh karena itu penulis memohon maaf dan menerima kritik terhadap apa yang telah penulis tuangkan dalam karya ilmiah skripsi ini.

Bandar Lampung, 22 Januari 2024

Penulis,

Aldo Wijaya

DAFTAR ISI

	Halaman
DAFTAR TABEL	viii
DAFTAR GAMBAR	x
I. PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	6
1.3 Tujuan Penelitian	6
1.4 Manfaat Penelitian	6
1.5 Batasan Masalah	7
II. TINJAUAN PUSTAKA	8
2.1 Dasar Teori	8
2.1.1 <i>Personal Extreme Programming (XP)</i>	8
2.1.2 Kuisisioner	10
2.1.3 Observasi.....	10
2.1.4 Kebutuhan	10
2.1.5 <i>User Stories</i>	11
2.1.7 Diagram.....	12
2.1.8 Pengujian.....	15
2.2 Penelitian Terdahulu	22
2.3 Analisis Kompetitor	25
2.4 Tools yang Digunakan	26
2.4.1 Visual Studio Code	26
2.4.2 Java Script	26
2.4.3 React JS	26
2.4.4 <i>Application Programming Interface (API)</i>	27
2.4.5 Firebase	27
III. METODOLOGI PENELITIAN	29

3.1 Waktu dan Tempat	29
3.1.1 Waktu	29
3.1.2 Tempat.....	29
3.2 Alat dan Bahan Penelitian	30
3.2.1 Alat Penelitian.....	30
3.2.2 Bahan Penelitian.....	31
3.3 Tahapan Penelitian	32
3.3.1 <i>Requirements</i>	32
3.3.2 <i>Planning</i>	33
3.3.3 <i>Iteration Initialization</i>	34
3.3.4 <i>Design</i>	34
3.3.5 <i>Implementation</i>	34
3.3.6 <i>System Testing</i>	35
3.3.7 <i>Retrospective</i>	35
IV. PEMBAHASAN	37
4.1 Requirements	37
4.1.1 Pengumpulan Data	37
4.1.2 Spesifikasi Kebutuhan.....	44
4.2 Planning	48
4.3 Iteration Initialization	49
4.4 Iterasi 1	51
4.4.1 <i>Design</i>	51
4.4.2 <i>Implementation</i>	52
4.4.3 <i>System Testing</i>	54
4.4.4 <i>Retrospective</i>	54
4.5 Iterasi 2	55
4.5.1 <i>Design</i>	55
4.5.2 <i>Implementation</i>	56
4.5.3 <i>System Testing</i>	59
4.5.4 <i>Retrospective</i>	60
4.6 Iterasi 3	61
4.6.1 <i>Design</i>	61

4.6.2	<i>Implementation</i>	61
4.6.3	<i>System Testing</i>	64
4.6.4	<i>Retrospective</i>	66
4.7	Iterasi 4	67
4.7.1	<i>Design</i>	67
4.7.2	<i>Implementation</i>	69
4.7.3	<i>System Testing</i>	72
4.7.4	<i>Retrospective</i>	75
4.8	Iterasi 5	76
4.8.1	<i>Design</i>	76
4.8.2	<i>Implementation</i>	78
4.8.3	<i>System Testing</i>	81
4.8.4	<i>Retrospective</i>	85
4.9	Iterasi 6	86
4.9.1	<i>Design</i>	86
4.9.2	<i>Implementation</i>	87
4.9.3	<i>System Testing</i>	90
4.9.4	<i>Retrospective</i>	95
4.10	Iterasi 7	103
4.10.1	<i>Design</i>	103
4.10.2	<i>Implementation</i>	105
4.10.3	<i>System Testing</i>	110
4.10.4	<i>Retrospective</i>	116
V.	PENUTUP	123
5.1	Kesimpulan	123
5.2	Saran	124
DAFTAR PUSTAKA		125
LAMPIRAN		129

DAFTAR TABEL

Tabel	Halaman
Tabel 1. Data pra survei (umum).....	2
Tabel 2. Data pra survei (penghafal).....	2
Tabel 3. Simbol dan fungsi flowchart	14
Tabel 4. Penelitian yang mengimplementasikan UAT	15
Tabel 5. Daftar pertanyaan SUS.....	18
Tabel 6. Waktu pelaksanaan penelitian	29
Tabel 7. Perangkat keras	30
Tabel 8. Perangkat lunak	30
Tabel 9. Data hasil kuisioner (umum).....	38
Tabel 10. Data hasil kuisioner (penghafal)	38
Tabel 11. Analisis SWOT kompetitor	43
Tabel 12. Kebutuhan fungsional	44
Tabel 13. Kebutuhan non fungsional	45
Tabel 14. <i>User stories</i>	45
Tabel 15. Rencana pelaksanaan iterasi.....	48
Tabel 16. Pelaksanaan inisialisasi iterasi	50
Tabel 17. <i>System testing</i> iterasi ke-1	54
Tabel 18. Evaluasi estimasi dan realisasi iterasi 1	54
Tabel 19. <i>System testing</i> iterasi ke-2	59
Tabel 20. Evaluasi estimasi dan realisasi iterasi 2	60
Tabel 21. <i>System testing</i> iterasi ke-3	64
Tabel 22. Evaluasi estimasi dan realisasi iterasi 3	66
Tabel 23. <i>System testing</i> iterasi ke-4	72
Tabel 24. Evaluasi estimasi dan realisasi iterasi 4	75
Tabel 25. <i>System testing</i> iterasi ke-5	81
Tabel 26. Evaluasi estimasi dan realisasi iterasi 5	85

Tabel 27. <i>System testing</i> iterasi ke-6	90
Tabel 28. Evaluasi estimasi dan realisasi iterasi 6	95
Tabel 29. Daftar nama responden SUS iterasi 6	96
Tabel 30. Rekapitulasi nilai kuisisioner	96
Tabel 31. Perhitungan skor menggunakan aturan SUS	97
Tabel 32. Hasil perhitungan skor (sebelum dikali 2,5)	97
Tabel 33. Jumlah skor	98
Tabel 34. Hasil akhir perhitungan skor	98
Tabel 35. Daftar evaluator pengujian HE	100
Tabel 36. Hasil rekapitulasi severity rating HE	100
Tabel 37. Rekomendasi evaluator HE	101
Tabel 38. <i>System testing</i> iterasi ke-7	110
Tabel 39. Daftar nama responden SUS	116
Tabel 40. Rekapitulasi nilai kuisisioner	117
Tabel 41. Perhitungan skor menggunakan aturan SUS	118
Tabel 42. Hasil perhitungan skor (sebelum dikali 2,5)	118
Tabel 43. Jumlah skor	119
Tabel 44. Hasil akhir perhitungan skor	119
Tabel 45. Daftar evaluator pengujian HE	121
Tabel 46. Hasil rekapitulasi <i>severity rating</i> HE	121
Tabel 47. Rekomendasi evaluator HE	122

DAFTAR GAMBAR

Gambar	Halaman
Gambar 1. Tahapan PXP [3].....	8
Gambar 2. Ilustrasi black box testing [4].	16
Gambar 3. Standar penilaian SUS [19].....	19
Gambar 4. Tahapan PXP [3].....	32
Gambar 5. Fitur riwayat muslim pro.....	40
Gambar 6. Fitur pencarian muslim pro.	41
Gambar 7. Search box muslim pro.....	41
Gambar 8. Hasil pencarian muslim pro.....	41
Gambar 9. Ulasan pengguna 1.	42
Gambar 10. Ulasan pengguna 2.	42
Gambar 11. Ulasan pengguna 3.	42
Gambar 12. Ulasan pengguna 4.	42
Gambar 13. Use case diagram sistem.	49
Gambar 14. Desain daftar surat.....	51
Gambar 15. Desain informasi dan audio surat.....	51
Gambar 16. Tampilan daftar surat.....	52
Gambar 17. Tampilan audio dan informasi surat.....	52
Gambar 18. Kode program fetch API daftar surat.....	53
Gambar 19. Desain informasi ayat (arab & arti) dan audio ayat.....	55
Gambar 20. Desain tafsir ayat.....	55
Gambar 21. Tampilan informasi dan audio ayat.....	56
Gambar 22. Tampilan tafsir ayat.....	57
Gambar 23. Kode program fetch API ayat.....	57
Gambar 24. Kode program fetch API tafsir.....	58
Gambar 25. Desain fitur cari, murojaah, dan setting.....	61
Gambar 26. Tampilan fitur cari.....	61

Gambar 27. Tampilan fitur murojaah	62
Gambar 28. Tampilan fitur setting huruf.....	63
Gambar 29. Desain halaman masuk.....	67
Gambar 30. Desain halaman daftar	67
Gambar 31. Desain fitur keluar	68
Gambar 32. Desain class diagram akun pengguna.....	68
Gambar 33. Tampilan halaman masuk.....	69
Gambar 34. Tampilan halaman daftar	69
Gambar 35. Tampilan fitur keluar	70
Gambar 36. Basis data autentikasi pengguna.....	71
Gambar 37. Desain halaman kontak	76
Gambar 38. Desain button simpan terakhir dibaca	76
Gambar 39. Desain fitur terakhir dibaca	77
Gambar 40. Desain class diagram terakhir baca	77
Gambar 41. Tampilan halaman kontak	78
Gambar 42. Tampilan button simpan terakhir dibaca	78
Gambar 43. Tampilan fitur simpan terakhir dibaca.....	79
Gambar 44. Basis data terakhir dibaca.....	80
Gambar 45. Desain riwayat hafalan	86
Gambar 46. Desain button simpan riwayat hafalan	86
Gambar 47. Desain class diagram riwayat hafalan	87
Gambar 48. Tampilan riwayat hafalan	87
Gambar 49. Tampilan button simpan riwayat hafalan	88
Gambar 50. Basis data riwayat hafalan.....	89
Gambar 51. Desain verifikasi aksi pengguna.....	103
Gambar 52. Desain fitur skip to content	103
Gambar 53. Fitur arrow up.....	104
Gambar 54. Desain halaman bantuan.....	104
Gambar 55. Desain rekomendasi riwayat email	105
Gambar 56. Tampilan fitur konfirmasi meninggalkan halaman	105
Gambar 57. Tampilan fitur konfirmasi keluar.....	106
Gambar 58. Tampilan fitur konfirmasi hapus hafalan.....	106

Gambar 59. Tampilan fitur skip to content	107
Gambar 60. Tampilan fitur arrow up.....	107
Gambar 61. Tampilan rekomendasi email yang pernah digunakan	108
Gambar 62. Tampilan informasi syarat password.....	108
Gambar 63. Tampilan fitur dokumentasi.....	109
Gambar 64. Tampilan status riwayat hafalan tidak ditemukan	109

I. PENDAHULUAN

1.1 Latar Belakang

Al-Qur'an merupakan kitab suci umat islam diseluruh dunia. Al-Qur'an adalah firman Allah Swt. yang diturunkan melalui Malaikat Jibril kepada Nabi Muhammad Saw. guna disampaikan kepada seluruh umat manusia di dunia. Al-Qur'an sebagai kitab suci di dalamnya terdapat 30 juz, 114 surat, dan 6000 ayat. Al-Qur'an sendiri dituliskan dengan menggunakan bahasa Arab. Isinya yang sejalan dengan berbagai fakta ilmiah dan mampu memberikan arahan dalam aspek kehidupan manusia, tidak hanya menjadikannya sebagai kitab suci, melainkan juga sebagai pedoman hidup umat islam diseluruh dunia.

Saat ini manusia sedang berada di era *society* 5.0. Era ini berfokus pada penerapan berbagai inovasi dan pengembangan teknologi yang distimulasi oleh era industri 4.0 sehingga mampu memecahkan masalah yang dihadapi oleh umat manusia. Era ini juga dapat diartikan sebagai suatu era dimana manusia menjadi pusat dari pengembangan dan inovasi teknologi, sehingga teknologi mampu meningkatkan kualitas hidup manusia [1]. Sederhananya, era ini dapat dipahami sebagai meningkatnya pemanfaatan teknologi informasi yang begitu signifikan guna mempermudah dan mendukung kehidupan manusia.

Pengaruh perkembangan teknologi informasi tidak hanya terbatas dalam hal pekerjaan, melainkan juga dalam hal keagamaan. Indonesia sebagai negara dengan jumlah penduduk beragama islam terbanyak di dunia, tentu harus mendukung teknologi informasi guna meningkatkan kualitas pelaksanaan dan pemahaman agama yang lebih baik lagi. Terdapat banyak sekali pemanfaatan teknologi informasi yang telah diimplementasikan untuk mempermudah aktivitas keagamaan masyarakat di dunia, salah satunya adalah Al-Qur'an digital.

Al-Qur'an digital merupakan salah satu bentuk pemanfaatan teknologi yang dapat meningkatkan kualitas pemahaman dan pembelajaran kitab suci Al-Qur'an

umat islam di dunia. Hadirnya Al-Qur'an digital memberikan kemudahan dari segi ruang dan waktu bagi umat islam yang hendak mempelajari dan mengamalkan perintah yang ada di dalam Al-Qur'an [2]. Kutipan tersebut menjelaskan bahwa Al-Qur'an digital membawa berbagai dampak positif bagi umat islam diberbagai wilayah dunia. Beberapa manfaat hadirnya Al-Qur'an digital di antaranya adalah peningkatan aksesibilitas Al-Qur'an bagi yang memiliki kendala dalam membawa Al-Qur'an mushaf, kemudahan dalam menyebarkan agama islam, kemudahan dalam membaca dan menghafal Al-Qur'an, menyediakan berbagai fitur yang dapat meningkatkan efektivitas pembelajaran dan pengamalan Al-Quran. Namun perlu diketahui bahwa pengembangan aplikasi Al-Qur'an digital ini tidak dilakukan untuk menggantikan Al-Qur'an mushaf, bagaimanapun Al-Qur'an dalam bentuk mushaf atau kitab tetap menjadi prioritas utama.

Tabel 1. Data pra survei (umum)

Pertanyaan	Iya	Tidak
Apakah Anda pernah menggunakan Al-Qur'an digital?	60	0
Apakah Anda memiliki Al-Qur'an digital?	52	8
Apakah Anda sering membawa gawai?	58	2
Apakah terdapat fitur pencatatan riwayat hafalan pada Al-Qur'an digital yang Anda gunakan?	16	44
Apakah terdapat fitur untuk murojaah (mengulang / menguji) pada Al-Qur'an digital yang Anda gunakan?	15	45
Apakah terdapat fitur pencatatan ayat yang terakhir dibaca pada Al-Qur'an digital yang Anda gunakan?	44	16
Apakah terdapat fitur pencarian berdasarkan teks, nomor, dan terjemahan ayat, pada Al-Qur'an digital yang Anda gunakan?	47	12

Tabel 2. Data pra survei (penghafal)

Pertanyaan	Iya	Tidak
Apakah Anda pernah / sedang menghafal Al-Qur'an?	15	0
Apakah Anda pernah menggunakan Al-Qur'an digital?	12	3
Apakah Anda memiliki Al-Qur'an digital?	10	5
Apakah Anda sering membawa gawai?	14	1

Apakah Al-Qur'an digital penting dalam mendukung proses belajar dan menghafal Al-Qur'an?	11	4
Apakah terdapat fitur pencatatan riwayat hafalan pada Al-Qur'an digital yang Anda gunakan?	2	13
Apakah terdapat fitur untuk murojaah (mengulang / menguji) pada Al-Qur'an digital yang Anda gunakan?	0	15
Apakah terdapat fitur pencatatan ayat yang terakhir dibaca pada Al-Qur'an digital yang Anda gunakan?	8	7
Apakah terdapat fitur pencarian berdasarkan teks, nomor, dan terjemahan ayat, pada Al-Qur'an digital yang Anda gunakan?	9	6

Tabel 1 merupakan data pra survei yang diperoleh dari 60 orang mahasiswa beragama islam yang ada di Universitas Lampung dengan berbagai program studi mulai dari angkatan 2020, 2021, dan 2022. Sedangkan tabel 2 merupakan data pra survei yang diperoleh dari 15 orang yang pernah atau sedang menghafal Al-Qur'an. Responden pada pada tabel 2 memiliki usia 21 – 22 tahun dan merupakan alumni dari sekolah-sekolah bernuansa islam seperti madrasah dan pondok pesantren.

Berdasarkan data pra survei pada tabel 1 dan tabel 2 tersebut, maka dapat disimpulkan bahwa hampir seluruh responden memiliki Al-Quran digital dan lebih sering membawa gawai daripada Al-Qur'an mushaf. Data tersebut juga menunjukkan hasil bahwa seluruh responden pernah menggunakan Al-Qur'an digital. Hal ini tentu merupakan salah bentuk pengaruh teknologi pada bidang keagamaan dan juga menjadi bukti bahwa pada saat ini manusia hidup berdampingan dengan teknologi. Namun berdasarkan data pra survei tersebut, maka dapat disimpulkan juga bahwasanya masih terdapat beberapa kekurangan pada aplikasi Al-Qur'an digital yang tersebar luas melalui internet. Mengacu pada jawaban responden pada kuisisioner yang telah dilaksanakan, masih ada aplikasi yang belum memiliki fitur pencatatan riwayat hafalan dan fitur *murojaah* untuk mengulang hafalan guna menguji ingatan terkait hafalan yang dimiliki. Jawaban reponden pada kuisisioner tersebut juga menyebutkan bahwa terdapat aplikasi Al-Qur'an digital yang masih belum memiliki fitur pencatatan ayat yang terakhir kali dibaca oleh pengguna. Disamping itu, masih terdapat aplikasi yang belum menyediakan fitur pencarian berdasarkan teks, nomor, dan terjemahan ayat.

Padahal beberapa fitur yang tidak disediakan oleh beberapa aplikasi Al-Qur'an digital tersebut merupakan fitur yang sangat fundamental dan dibutuhkan oleh pengguna dalam meningkatkan kualitas pembelajaran Al-Qur'an.

Pada penelitian ini dipilih salah satu aplikasi sejenis yang selanjutnya akan digunakan dan dibahas pada analisis kompetitor, aplikasi tersebut bernama Muslim Pro. Meskipun aplikasi Muslim Pro tidak dalam bentuk aplikasi web sebagaimana aplikasi Al-Qur'an digital yang dikembangkan dalam penelitian ini, namun Muslim Pro sudah dapat dikatakan mewakili keseluruhan aplikasi Al-Qur'an digital yang tersebar di dunia internet, baik dalam bentuk aplikasi web, desktop, maupun mobile. Hal ini disebabkan karena kelengkapan fitur yang ada dalam aplikasi tersebut. Alasan lain dari pemilihan aplikasi ini sebagai kompetitor adalah karena aplikasi Muslim Pro merupakan aplikasi yang paling banyak digunakan oleh responden berdasarkan data pra survei yang telah dilakukan sebelumnya. Dimana berdasarkan data pra survei tersebut, aplikasi Muslim pro memiliki pengguna terbanyak, yaitu 39 pengguna dari 75 responden. Kemudian aplikasi ini juga memiliki riwayat unduhan sebanyak 100 juta kali di play store android. Meskipun begitu, berdasarkan ulasan pengguna dan hasil analisis secara sepintas, masih terdapat beberapa kekurangan pada aplikasi ini jika dilihat dari beberapa aspek.

Beberapa penelitian terdahulu yang relevan telah mampu menciptakan suatu produk berupa Al-Qur'an digital yang dapat menjadi salah satu alternatif bagi umat muslim untuk membaca dan menghafalkan Al-Qur'an. Meskipun penelitian tersebut telah berhasil menghasilkan suatu produk berupa Al-Qur'an digital, masih ada beberapa kekurangan yang ada dalam produk tersebut, misalnya berupa fitur yang masih kurang lengkap dalam Al-Qur'an digital tersebut.

Berbagai faktor yang telah dijelaskan sebelumnya, mulai dari data hasil pra survey hingga penelitian terdahulu, menjadi alasan mendasar pentingnya pelaksanaan penelitian ini. Oleh karena itu, penelitian ini akan berfokus pada pengembangan fitur-fitur penting yang disesuaikan dengan kebutuhan pengguna untuk memberikan pengalaman pengguna yang lebih baik. Pada penelitian ini terdapat 22 kebutuhan fungsional yang telah disesuaikan dengan kebutuhan pengguna. Kebutuhan fungsional tersebut dibuat berdasarkan hasil data pra survei dan observasi terhadap aplikasi sejenis yang relevan.

Pada penelitian ini digunakan metode *Personal Extreme Programming* (PXP) dalam hal pengembangan perangkat lunak. Alasan penggunaan metode ini adalah untuk mempermudah pengerjaan pengembangan perangkat lunak yang dikerjakan oleh satu orang saja, hal ini disebabkan karena metode ini mengurangi upaya dokumentasi dan pemeliharaan, namun tetap menjaga prinsip dasar pengembangan [3].

Dalam hal pengujian sistem, penelitian ini akan menggunakan 3 pengujian utama, yaitu *black box testing*, *System Usability Scale* (SUS), dan *Heuristic Evaluation* (HE). Keuntungan dari penggunaan metode *black box testing* di antaranya adalah pengujian yang dilakukan berdasarkan sudut pandang pengguna aplikasi, sehingga pengujian dapat dilakukan oleh orang yang tidak memahami struktur internal perangkat lunak tersebut, atau bahkan orang yang tidak memahami pemrograman sekalipun [4]. Alasan penggunaan SUS di antaranya adalah skala pengujian yang relatif lebih mudah dimengerti oleh responden yang merupakan orang awam, dapat dilakukan dengan jumlah sampel yang kecil, memiliki hasil yang efektif [5]. Selanjutnya keuntungan dari penggunaan metode pengujian *heuristic evaluation* di antaranya adalah biaya yang murah, tidak memerlukan perencanaan yang kompleks, dapat digunakan dalam proses pengembangan, dan pengujian berdasarkan sudut pandang penguji atau bersifat subjektif [6].

1.2 Rumusan Masalah

1. Bagaimana cara mengembangkan aplikasi Al-Qur'an digital yang dapat memfasilitasi pembaca dan penghafal Al-Qur'an?
2. Apakah Al-Quran digital yang dikembangkan sudah sesuai dengan kebutuhan pengguna?
3. Apakah Al-Quran digital yang dikembangkan sudah memiliki nilai *usability* atau kebergunaan yang dapat diterima (*acceptable*) oleh pengguna?

1.3 Tujuan Penelitian

Adapun tujuan dari penelitian ini di antaranya adalah:

1. Mengembangkan Al-Quran digital yang dapat memfasilitasi pembaca dan penghafal Al-Qur'an.
2. Menguji fungsionalitas aplikasi Al-Quran digital agar sesuai dengan kebutuhan pengguna.
3. Menguji nilai *usability* atau kebergunaan aplikasi Al-Quran digital agar dapat diterima oleh pengguna.

1.4 Manfaat Penelitian

Adapun manfaat dari penelitian ini di antaranya adalah:

1. Penelitian ini memberikan kontribusi dalam mendukung kemajuan teknologi melalui pengembangan aplikasi Al-Qur'an digital.
2. Penelitian ini dapat menjadi pijakan bagi penelitian lanjutan di bidang yang relevan.
3. Penelitian ini mendukung dalam usaha peningkatan pemahaman dan pengetahuan Al-Qur'an bagi masyarakat.
4. Penelitian ini dapat menjadi sumber referensi yang terpercaya bagi pembaca.

1.5 Batasan Masalah

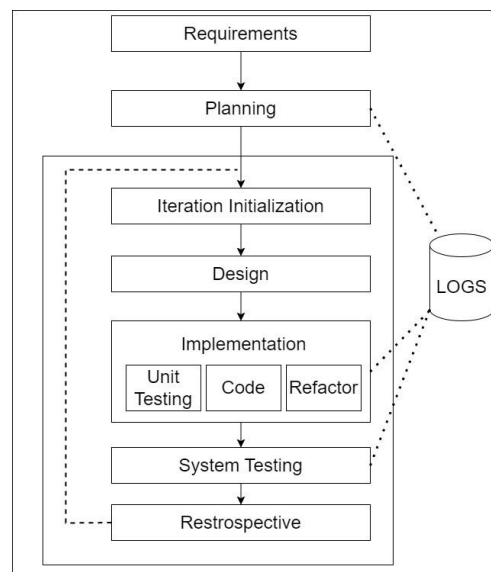
Adapun batasan masalah pada penelitian ini di antaranya adalah:

1. Pengembangan aplikasi Al-Qur'an digital pada penelitian ini memanfaatkan API yang dikembangkan oleh pihak ketiga (EQuran.id). API ini dikembangkan oleh komunitas yang bernama Muslim Dev. Dalam dokumentasi resmi API tersebut, tertulis bahwa sumber utama dari data yang terdapat pada API tersebut berasal dari Aplikasi Quran Kementerian Agama Republik Indonesia. Selanjutnya untuk audio murottal didapat dari Quran central dan Verse By Verse Quran. Sehingga kesalahan data Al-Quran di dalamnya tidak menjadi tanggung jawab pengembang aplikasi Al-Qur'an digital dalam penelitian ini.
2. Penelitian ini memanfaatkan aplikasi firebase untuk autentikasi pengguna dan penyimpanan data, sehingga penelitian ini menerima dan mengikuti syarat-syarat yang ditetapkan oleh firebase.
3. Penelitian ini lebih banyak berfokus pada pengembangan aplikasi dari sisi *frontend*, sehingga aktivitas *backend* di dalamnya lebih kearah pemahaman konseptual bukan teknikal.

II. TINJAUAN PUSTAKA

2.1 Dasar Teori

2.1.1 *Personal Extreme Programming (XP)*



Gambar 1. Tahapan PXP [3].

Personal Extreme Programming (XP) merupakan salah satu metodologi pengembangan perangkat lunak yang bertujuan untuk meningkatkan kinerja dari pengembang perangkat lunak yang bekerja secara individual. PXP sendiri merupakan salah satu metode agile yang dikembangkan dari hasil kombinasi antara *Personal Software Process (PSP)* dan *Extreme Programming (XP)*. PSP merupakan metode pengembangan perangkat lunak yang secara spesifik dibuat untuk pengembangan perangkat lunak yang dikerjakan oleh individu. Namun, dalam proses pengimplementasiannya, perlu pengetahuan yang cukup baik serta usaha yang besar dalam hal pemeliharaan dan pembuatan dokumentasinya. Berdasarkan hal tersebut, maka PXP dikembangkan dengan menerapkan *subset* dari XP namun tetap mempertahankan prinsip-prinsip dasar pengembangan yang ada pada PSP, serta meminimalisir jumlah dokumentasi dan upaya pemeliharaan. PXP juga dibuat

dengan tujuan untuk meminimalisir waktu yang dihabiskan dalam pengembangan perangkat lunak, karena apabila interval pengiriman tertunda, maka akan mengurangi keuntungan kompetitif di pasar [3].

Proses pengembangan perangkat lunak dengan menggunakan metode *Personal Extreme Programming* (PXP) bersifat iteratif, sehingga memungkinkan pengguna (pengembang) lebih fleksibel ketika terjadi perubahan-perubahan tertentu pada saat pelaksanaan proses pengembangan. Gambar 1 merupakan 7 tahapan yang perlu dilakukan pada implementasi metode PXP saat pengembangan perangkat lunak. Tujuh tahapan tersebut di antaranya adalah *requirements*, *planning*, *iteration initialization*, *design*, *implementation*, *system testing*, dan *retrospective*. Tahap *requirements* merupakan tahap pembuatan dokumen yang berisi kebutuhan fungsional dan non-fungsional untuk sistem yang yang dibuat, serta dilanjutkan dengan pembuatan *user stories*. Kemudian pada tahap *planning*, pengembang menyusun daftar tugas yang akan dikerjakan berdasarkan *user stories* yang telah dibuat sebelumnya dengan membaginya menjadi beberapa iterasi. Pada tahap ini juga direncanakan bahasa pemrograman, kerangka, dan *tools* yang akan digunakan. Kemudian tahap *iteration initialization* merupakan awal setiap iterasi yang dilakukan, dimana pengembang memilih tugas yang akan menjadi fokus pada iterasi tersebut. Selanjutnya pada tahap *design*, pengembang memodelkan perangkat lunak berdasarkan tugas-tugas yang telah disusun sebelumnya. Tahap implementasi merupakan tahap pengerjaan kode program, di dalamnya terbagi atas 3 sub proses, yaitu *unit testing*, *code*, dan *refactor*. Selanjutnya seluruh fitur yang telah dikerjakan, dilakukan *testing* untuk menguji seluruh fitur tersebut. Kemudian segala kesalahan yang ada menjadi bahan evaluasi di *retrospective* dan diperbaiki kembali melalui tahap *iteration initialization* [3].

Sebuah studi pengembangan sistem informasi aktivitas pelatihan kerja dengan menggunakan metode PXP juga menyimpulkan bahwa sistem yang dikembangkan dengan menggunakan metode tersebut telah memenuhi dan sesuai dengan kebutuhan pengguna. Disamping itu, penggunaan metode PXP dalam pengembangan sistem informasi tersebut telah mempermudah pengguna dalam mengembangkan program yang sesuai dengan *user stories* hanya dalam waktu yang relatif singkat [7].

2.1.2 Kuisisioner

Kuisisioner merupakan sekumpulan daftar pertanyaan yang jawabannya akan diisi secara langsung oleh responden. Pada implementasi metode kuisisioner, responden membaca kuisisioner, menginterpretasikan apa yang diharapkan, dan menuliskan jawabannya. Karena dalam proses pelaksanaan kuisisioner tidak ada orang yang menjelaskan arti dari setiap pertanyaan, maka pertanyaan yang ada pada kuisisioner perlu dibuat dengan jelas dan mudah dimengerti. Tata letak kuisisioner juga perlu disusun sedemikian rupa sehingga memudahkan responden dalam membaca dan memahami maksud setiap pertanyaan. Kuisisioner harus dikembangkan dengan gaya yang interaktif, agar pembaca dapat merasakan seolah-olah sedang berbicara atau berinteraksi secara langsung. Pada dasarnya tidak terdapat aturan baku yang mengatur tentang batasan jumlah responden yang digunakan dalam kuisisioner. Dalam penelitian kuantitatif, semakin banyak responden yang terlibat, maka semakin akurat data yang akan dihasilkan [8].

2.1.3 Observasi

Kata “observasi” berasal dari bahasa latin yang memiliki arti melihat dan memperhatikan. Observasi merupakan proses melihat, mengamati, dan meninjau suatu objek dengan teliti. Metode observasi merupakan salah satu metode pengumpulan data yang ada dan validitasnya dapat dipertanggungjawabkan. Kelebihan dari penggunaan metode observasi dalam pengumpulan data salah satunya adalah dapat meminimalisir pengeluaran biaya [9].

2.1.4 Kebutuhan

1. Kebutuhan Fungsional

Kebutuhan fungsional pada perangkat lunak dapat didefinisikan sebagai hal-hal apa saja yang sebaiknya dapat dilakukan oleh sistem. Kebutuhan sistem ini mengacu pada harapan pengguna terhadap *software* yang akan dikembangkan dan pendekatan umum yang digunakan oleh penulis ketika menulis kebutuhan sistem. Kebutuhan fungsional mendefinisikan spesifikasi fasilitas yang harus disediakan oleh sistem. Dalam prinsipnya, spesifikasi kebutuhan fungsional dalam suatu sistem

harus dibuat lengkap dan konsisten. Lengkap dalam hal ini berarti semua yang dibutuhkan oleh pengguna harus didefinisikan. Sedangkan konsisten dapat diartikan sebagai kebutuhan fungsional sebaiknya tidak boleh bertentangan antara yang satu dengan yang lainnya. Namun dalam praktiknya, terkhusus sistem yang besar dan kompleks, sering kali kebutuhan fungsional yang didefinisikan tidak lengkap dan konsisten. Hal ini bisa terjadi karena beberapa faktor, di antaranya adalah potensi yang besar untuk menulis kesalahan dan kelalaian saat menulis spesifikasi sistem yang kompleks, banyaknya *stakeholder* yang terkait dengan pengembangan sistem seringkali memiliki perbedaan pendapat dalam menentukan kebutuhan sistem [10].

2. Kebutuhan Non Fungsional

Kebutuhan non fungsional merupakan kebutuhan yang berhubungan dengan kemampuan suatu sistem, seperti keandalan, waktu respons, dan sebagainya. Sebagai kebutuhan non fungsional juga dapat dipahami sebagai kebutuhan yang tidak berkaitan langsung dengan fungsi atau layanan spesifik yang diberikan oleh pengguna kepada sistem. Kebutuhan non fungsional dapat menentukan batasan pada implementasi sistem seperti kemampuan perangkat I/O. Beberapa contoh kebutuhan non fungsional di antaranya adalah kinerja, keamanan, ketersediaan, atau batasan karakteristik sistem secara keseluruhan. Kebutuhan non fungsional dapat berpengaruh terhadap arsitektur keseluruhan sistem. Satu kebutuhan non fungsional dalam suatu sistem dapat mempengaruhi atau menghasilkan kebutuhan fungsional baru yang relevan. Kebutuhan non fungsional dapat muncul dari berbagai karakteristik sistem ataupun sumber eksternal [10].

2.1.5 *User Stories*

User Stories (US) merupakan salah satu kunci utama yang berfungsi untuk menentukan kebutuhan sistem pada metodologi *agile*. *User stories* merupakan dokumen yang menjelaskan kebutuhan fungsional pengguna mulai dari tingkatan yang paling rendah. Dalam penggunaannya, *User Stories* memiliki format penulisan yang telah dikembangkan oleh para ahli. Format tersebut dapat ditulis dengan sebagai [siapa], saya ingin [apa], sehingga [karena]. Sederhananya format tersebut dapat dipahami sebagai siapa aktor yang akan menggunakan sistem,

fungsionalitas apa yang diinginkan oleh aktor tersebut, dan alasan mengapa fungsionalitas tersebut dibutuhkan oleh aktor [11].

2.1.6 Moscow Prioritization

Moscow prioritization merupakan salah satu metode yang dapat digunakan untuk menentukan prioritas pengembangan fitur dalam proses *System Development Life Cycle* (SDLC). Moscow prioritization ini awalnya diimplementasikan dalam salah satu metode pengembangan sistem yaitu DSDM (*Dynamic Systems Development Method*). Moscow juga merupakan akronim dari aturan Moscow itu sendiri, yaitu *must have*, *should have*, *could have*, dan *won't have*. *Must have* berarti bahwa fitur tersebut merupakan fitur yang sangat penting dan tidak dapat ditinggalkan oleh sistem. *Should have* merupakan penanda bahwa fitur tersebut merupakan salah satu fitur penting yang sebaiknya ada, namun keberhasilan proyek tidak ditentukan oleh fitur ini. Fitur *should have* dapat ditinggalkan dalam rilis jika terdapat kendala seperti keterbatasan waktu. *Could have* berarti bahwa fitur tersebut merupakan fitur yang dapat ada, namun jika ditinggalkan tidak akan memberikan dampak apapun. *Won't have* berarti bahwa fitur tersebut dapat ditinggalkan, namun dapat diimplementasikan jika terdapat kelebihan anggaran atau waktu [12].

2.1.7 Diagram

1. Use Case Diagram

Use case diagram merupakan diagram yang memungkinkan pengguna untuk menggambarkan kemungkinan skenario penggunaan suatu sistem. Diagram ini menjelaskan bagaimana cara penggunaan suatu sistem, namun tidak membahas detail apapun seperti struktur data, basis data, algoritma, dsb. Diagram *use case* merupakan salah satu konsep fundamental dalam pengembangan berorientasi objek. Diagram ini digunakan selama proses desain dan analisis pada saat pengembangan sistem. *Use case* mewakili apa yang ingin dilakukan oleh pengguna. Sederhananya, *use case diagram* menjelaskan terkait dengan sistem, siapa saja yang berinteraksi dengan sistem, dan apa saja yang bisa dilakukan oleh aktor sebagai pengguna sistem tersebut. Adapun komponen penyusun diagram *use case* terdiri atas *use cases*, aktor, asosiasi, dan relasi antar *use cases*. Sebuah *use case*

mendeskripsikan fungsionalitas yang diharapkan dapat tersedia pada sistem yang dikembangkan. Aktor mendeskripsikan pengguna yang berinteraksi dengan sistem. Asosiasi digunakan untuk menghubungkan pengguna atau para aktor dengan *use cases*. Adapun lambang dari asosiasi adalah garis lurus tanpa adanya detail apapun. Relasi antar *use cases* merupakan penggambaran hubungan antar *use cases*. Relasi antar *use case* terbagi atas 2 jenis, yaitu *include* dan *extend*, keduanya dilambangkan dengan garis putus-putus dengan panah. Jika *use case A includes use case B*, maka dapat diartikan bahwa untuk menjalankan fungsionalitas *use case A*, perlu adanya aksi pada *use case B* (*use case B* harus sudah dieksekusi). Kemudian jika *use case B extend dengan use case A*, maka *use case B* dapat dijalankan oleh *use case A*, namun menjalankan *use case B* tidaklah wajib [13].

2. Class Diagram

Class diagram merupakan suatu diagram yang digunakan untuk memodelkan struktur dalam sistem yang sifatnya statis. Hal ini menjadikan *class diagram* tersusun untuk menggambarkan berbagai elemen-elemen dalam sistem serta hubungan antar elemen tersebut. *Class diagram* sendiri merupakan salah satu diagram *Unified Modelling Language* yang paling banyak digunakan. Dalam konsep pemrograman berorientasi objek, *class diagram* memvisualisasikan kelas-kelas yang ada di dalam sistem serta hubungan antar kelas tersebut. Dalam praktiknya, *class diagram* juga sering digunakan untuk keperluan dokumentasi. Dalam *class diagram*, kelas direpresentasikan oleh bentuk kotak yang di dalamnya terbagi atas 3 bagian. Bagian pertama diisi dengan nama kelas, umumnya nama kelas ditulis dengan huruf kapital, tebal, dan diposisikan di Tengah (*center*). Bagian kedua berisi berbagai atribut dari kelas tersebut. Selanjutnya bagian ketiga berisi operasi, fungsi, ataupun metode yang dapat dijalankan oleh kelas tersebut. Bagian kedua dan ketiga harus ditulis rata kiri dan penulisannya bersifat opsional, artinya bagian kedua dan ketiga bisa saja tidak dituliskan dalam suatu kelas. Dalam penulisannya, atribut dan fungsi dalam suatu kelas memiliki simbol-simbol untuk mendeskripsikan visibilitasnya. Simbol *public* (+) mendeskripsikan bahwa atribut dapat diakses oleh berbagai objek dan class manapun. Simbol *private* (-) mendeskripsikan bahwa atribut dapat diakses oleh kelas itu sendiri. Simbol

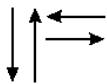

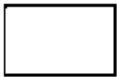
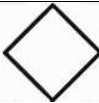

protected (#) mendeskripsikan bahwa atribut dapat diakses oleh kelas itu sendiri dan kelas turunannya. Selanjutnya kelas-kelas yang ada dalam *class diagram* dihubungkan dengan permodelan asosiasi. Hubungan antar kelas menggunakan asosiasi, seringkali menggunakan *many* (*), *one* (1), dan *zero* (0) untuk merepresentasikan nilai atau jumlah objeknya [13].

3. Diagram Alir

Diagram alir atau yang biasa disebut *Flowchart* merupakan diagram yang merepresentasikan langkah-langkah urutan suatu algoritma. Tujuan utama dari *flowchart* adalah untuk menemukan atau menciptakan urutan Langkah-langkah yang menunjukkan solusi atas suatu masalah. Terdapat beberapa aturan yang harus dipatuhi dalam pembuatan *flowchart*, di antaranya adalah simbol yang digunakan pada *flowchart* adalah simbol standar, logika program harus dibuat dari atas ke bawah atau dari kiri ke kanan, dan semua simbol *decision* harus dilabeli dengan baik dan benar. *Flowchart* juga dapat digunakan untuk menunjukkan urutan Langkah-langkah untuk melakukan suatu kegiatan atau pekerjaan [14].

Tabel 3 di bawah ini merupakan beberapa simbol *flowchart* yang paling sering digunakan, khususnya dalam penelitian ini.

Tabel 3. Simbol dan fungsi flowchart

NO	Simbol	Nama	Kegunaan
1		Arus	Penghubung antara simbol yang satu dengan simbol yang lainnya.
2		Terminator	Untuk menandakan awal (<i>start</i>) dan akhir (<i>end</i>) suatu diagram alir.
3		Proses	Menunjukkan proses atau tahapan yang akan dikerjakan.
4		Decision	Menunjukkan kondisi tertentu yang dihadapkan pada dua pilihan, yaitu <i>yes</i> atau <i>no</i> .
5		Connector	Menghubungkan antara 2 proses yang berada pada halaman yang sama.

2.1.8 Pengujian

1. User Acceptance Test (UAT)

User Acceptance Test (UAT) merupakan salah satu metodologi pengujian kebutuhan fungsional yang dapat dilakukan dengan menggunakan *black box testing*. Pengujian dengan metode UAT digunakan untuk mengetahui apakah produk yang dikembangkan sudah memenuhi kebutuhan pengguna. Secara umum, pengujian UAT terbagi atas 2 jenis mekanisme pelaksanaan, yaitu *alpha testing* menggunakan metode *black box* dan *beta testing* menggunakan *likert scale*. *Alpha testing* memiliki fokus dalam hal pengujian sistem untuk mengetahui apakah fungsi dari aplikasi sudah berjalan dengan benar sesuai dengan yang diharapkan dan terbebas dari *error* [15].

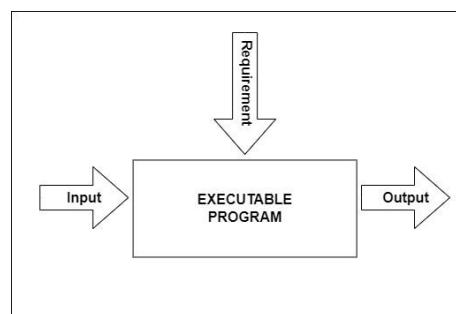
Tabel 4 merupakan beberapa penelitian yang mengimplementasikan metode UAT dalam pengembangan software menggunakan metodologi *Personal Extreme Programming (PXP)*:

Tabel 4. Penelitian yang mengimplementasikan UAT

Judul	Penulis	Tahun
Web-Based Information System for Job Training Activities Using Personal Extreme Programming (PXP)	S A Asri, I G A M Sunaya, E Rudiastari, W Setiawan	2017
Implementasi Metode Personal Extreme Programming dalam Pengembangan Sistem Manajemen Transaksi Perusahaan	Muhammad Ulfi, Gita Indah Marthasari, Ilyas Nuryasin	2020
Pengembangan Sistem Pengelolaan Peminjaman Auditorium Universitas Muhammadiyah Malang Berbasis Web Menggunakan Metode Personal Extreme Programming	Dana Heris Ardianzah, Ilyas Nuryasin, Briansyah Setio Wiyono	2022
Rancang Bangun Sistem Informasi Buku Tamu pada Dinas Pemuda, Olahraga, dan Pariwisata Kota Balikpapan dengan Metode Personal Extreme Programming	Ahmad Maulana Fikri, I Putu Deny Arthawan S P	2020
Sistem Informasi Program Keluarga Harapan (PKH) menggunakan Metode Personal Extreme Programming dengan Metode Prioritas Ranking	Abdullah Faqih Septiyanto, Wildan Suharso, Ilyas Nuryasin	2020

Aplikasi Eresha Mobile Berbasis Android dengan Metode Personal Extreme Programming di Era Industri 4.0	Agus Suharto, Mardiana	2020
Pengembangan Aplikasi Android untuk Monitoring Akademik Mahasiswa di Institut Teknologi Kalimantan dengan Metode Personal Extreme Programming	M G L Putra, M I Firdaus, M I Ramadhan, S Ardiansyah	2022
Sistem Informasi Parenting menggunakan Metode Design Thinking dan Personal Extreme Programming	M Awaluddin, R Afwani, B Irmawati	2022
Pengembangan Sistem Informasi Pemesanan Barang pada Distributor Cat Menggunakan Metode Pengembangan Personal Extreme Programming	A Wicaksana, S A Wicaksono, W Purnomo	2019

2. Black Box Testing



Gambar 2. Ilustrasi black box testing [4].

Black box testing merupakan salah satu metode pengujian perangkat lunak. Metode pengujian ini digunakan untuk menguji fungsionalitas pada perangkat lunak. Metode *black box testing* sendiri melakukan pengujian berdasarkan spesifikasi kebutuhan perangkat lunak. Pada dasarnya, *black box testing* memfokuskan pengujian terhadap masukan dan keluaran pada perangkat lunak sebagaimana yang divisualisasikan oleh gambar 2. Dimana suatu masukan yang dilakukan, diharapkan dapat memberikan keluaran yang sesuai dengan fungsional pada fitur perangkat lunak yang diuji. Metode ini mengacu pada *software requirement specification (SRS)*. Keuntungan dari penggunaan metode ini di antaranya adalah pengujian yang dilakukan berdasarkan sudut pandang pengguna aplikasi, sehingga pengujian dapat dilakukan oleh orang yang tidak memahami struktur internal perangkat lunak tersebut, atau bahkan orang yang tidak memahami pemrograman sekalipun [4].

Pengujian perangkat lunak dengan menggunakan metode *black box testing* dapat digunakan sepanjang siklus pengembangan dan pengujian perangkat lunak. Pengujian perangkat lunak yang dapat menggunakan *black box testing* di antaranya adalah *regression testing*, *unit testing*, *integration testing*, dan *system testing* [16].

3. System Usability Scale (SUS)

Usability atau kebergunaan bukan merupakan suatu hal yang sifatnya absolut atau mutlak. Pada umumnya, menentukan tingkat *usability* atau kebergunaan dari suatu sistem merupakan suatu hal yang tidak mungkin jika tidak dilakukan tanpa mendefinisikasi pengguna sistem tersebut, tugas dari pengguna, dan karakteristik fisik, organisasi, serta lingkungan tempat suatu sistem akan digunakan. Konsekuensi dari penggunaan pengukuran *usability* adalah bahwa sangat sulit untuk membandingkan nilai *usability* antara suatu sistem dengan sistem lainnya. Membandingkan nilai *usability* antara suatu sistem dengan sistem lainnya merupakan contoh nyata dari membandingkan dua hal yang berbeda, seperti membandingkan antara buah apel dan jeruk (keduanya memiliki konteks yang berbeda) [17].

System Usability Scale (SUS) merupakan suatu metode pengujian untuk menentukan nilai *usability* atau kebergunaan dari suatu sistem menggunakan sepuluh pertanyaan subjektif. Pada saat awal pengembangan, jumlah pertanyaan yang disusun pada metode SUS adalah sebanyak 50 pertanyaan. Terdapat 20 orang yang berasal dari kelompok *system engineering*, mulai dari sekertaris hingga *programmer* melakukan penilaian terhadap suatu sistem menggunakan 50 pertanyaan yang telah disusun sebelumnya menggunakan skala poin dari 1 – 5, mulai dari sangat tidak setuju hingga sangat setuju. Kemudian pertanyaan-pertanyaan yang mendapatkan respon sangat setuju atau sangat tidak setuju, dipilih sebagai pertanyaan yang akan dicantumkan dalam metode SUS, sehingga tersisa 10 pertanyaan terbaik. Hal ini dilakukan untuk menghindari adanya keraguan tanggapan yang dilakukan oleh responden. Kemudian pertanyaan pada metode SUS juga disusun secara bergantian, antara pertanyaan yang sifatnya positif dan negative. Tujuannya adalah agar responden membaca setiap pertanyaan dengan teliti [17].

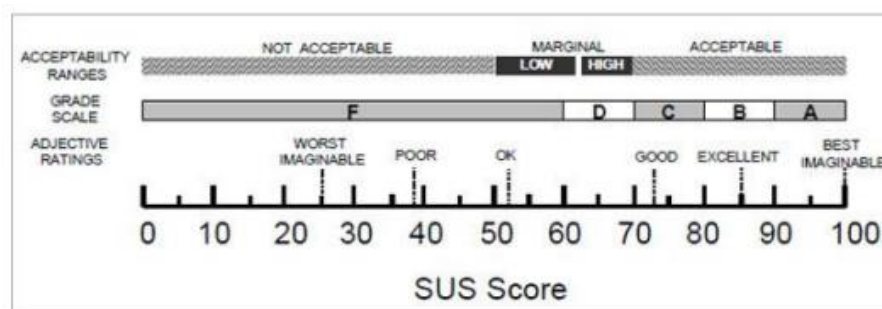
Tabel 5. Daftar pertanyaan SUS

NO	Pertanyaan	1	2	3	4	5
1	Saya pikir saya akan menggunakan sistem ini lebih sering					
2	Saya merasa sistem ini tidak perlu dibuat rumit seperti ini					
3	Saya merasa sistem ini mudah untuk digunakan					
4	Saya pikir saya membutuhkan bantuan dari seorang teknis untuk dapat menggunakan sistem ini					
5	Saya menemukan berbagai fitur dalam aplikasi terintegrasi dengan baik					
6	Saya merasa banyak hal yang tidak konsisten dalam sistem					
7	Saya membayangkan kebanyakan orang dapat dengan mudah menjalankan sistem ini					
8	Saya melihat sistem ini sangat rumit untuk digunakan					
9	Saya merasa sangat percaya diri dalam menggunakan sistem ini					
10	Saya perlu belajar banyak hal untuk sebelum saya dapat menggunakan aplikasi ini					

Metode SUS dapat digunakan setelah responden menggunakan sistem yang akan dievaluasi, tetapi sebelum pembekalan atau diskusi apapun dilakukan. Responden harus mengisi setiap pertanyaan yang ada secara langsung daripada harus memikirkan pertanyaan dalam waktu yang lama. Semua pertanyaan harus diisi, apabila responden memiliki keraguan untuk mengisi suatu pertanyaan, maka dia dapat memilih nilai tengah, yaitu poin 3. Dalam proses perhitungan nilai SUS, terdapat beberapa aturan yang menjadi tata cara dalam proses perhitungan nilainya. Aturan-aturan tersebut di antaranya adalah hasil penilaian setiap jawaban memiliki interval 0 – 4 dengan nilai 4 merupakan jawaban yang terbaik, pertanyaan dengan nomor ganjil (1, 3, 5, 7, 9) memiliki nilai skala jawaban dikurangi dengan 1, pertanyaan dengan nomor genap (2, 4, 6, 8, 10) maka perhitungannya adalah 5 dikurangi dengan skala jawaban responden, selanjutnya adalah melakukan penjumlahan jawaban setiap responden kemudian dikali dengan angka 2,5, Selanjutnya adalah menghitung nilai rata-rata jawaban pengujian yang berasal dari semua responden [17].

Adapun beberapa kelebihan dari penggunaan metode SUS di antaranya adalah skala pengujian yang relatif lebih mudah dimengerti oleh responden yang merupakan orang awam, dapat dilakukan dengan jumlah sampel yang kecil, memiliki hasil yang efektif. Dalam hal jumlah responden, SUS tidak memiliki konsep atau aturan baku yang mengatur tentang standar jumlah responden yang digunakan dalam pengujian. Ketentuan jumlah responden yang terlibat dalam pelaksanaan SUS merupakan kebebasan dan independensi dari peneliti [18]. Penelitian yang dilakukan oleh Tullis dan Stetson pada tahun 2004 juga menyebutkan bahwa SUS memungkinkan penggunaan jumlah responden yang kecil (antara 8 – 13) dan menghasilkan penilaian yang efektif [5].

SUS memiliki tiga sudut pandang dalam menentukan hasil perhitungan penilaian, di antaranya adalah *acceptability range*, *grade scale*, dan *adjective ratings*. *Acceptability* sendiri digunakan untuk melihat tingkat atau nilai penerimaan pengguna akhir terhadap suatu sistem. Berdasarkan sudut pandang *acceptability*, penilaian terbagi lagi atas *not acceptable*, *marginal*, dan *acceptable*. Adapun predikat *not acceptable* didapat jika skor perhitungan akhir SUS adalah 0 – 50, predikat *marginal low* didapat jika perhitungan antara 50 – 60, predikat *marginal high* di dapat jika skor 60 – 70, dan perhitungan *acceptable* di dapat jika nilainya adalah 70 – 100 [19]. Adapun penilaian SUS secara lebih detail dapat dilihat pada gambar 3 di bawah ini.



Gambar 3. Standar penilaian SUS [19]

4. Heuristic Evaluation (HE)

Heuristic Evaluation (HE) merupakan salah satu metode yang menguji tingkat *usability* atau kebergunaan dari suatu aplikasi. *Heuristic evaluation* juga dapat didefinisikan sebagai pengujian suatu sistem yang pada dasarnya dapat dikerjakan hanya dengan melihat tampilan antarmuka pengguna dan evaluator akan menyampaikan pendapatnya tentang apa yang baik dan buruk dari antarmuka tersebut. Keuntungan dari penggunaan metode pengujian *heuristic evaluation* di antaranya adalah biaya yang murah, tidak memerlukan perencanaan yang kompleks, dapat digunakan dalam proses pengembangan, dan pengujian berdasarkan sudut pandang penguji atau bersifat subjektif. Dalam hal penentuan kuantitas jumlah evaluator, maka tiga hingga lima evaluator merupakan rekomendasi terbaik pada pengujian *heuristic evaluation* [6]. Meskipun pengujian ini dapat dijalankan oleh siapapun, namun evaluator yang *expert* dan memiliki pengalaman tentu menjadi rekomendasi terbaik dibandingkan evaluator pemula. Evaluator *expert* merupakan seseorang yang telah memiliki gelar kelulusan atau seseorang yang memiliki pengalaman dalam bidang terkait [20].

Setiap evaluator menjalankan proses pengujian *heuristic evaluation* secara individual dan hanya diizinkan untuk berkomunikasi satu dengan yang lainnya apabila pengujian dan penilaian sudah selesai dilakukan. Hal ini menjadi hal yang sangat penting untuk menghindari pengaruh evaluator yang satu terhadap evaluator lainnya. Adapun *output* yang dihasilkan dari pengujian dengan menggunakan metode ini di antaranya adalah daftar masalah *usability* pada antarmuka pengguna beserta referensi prinsip yang digunakan. *Heuristic evaluation* tidak memberikan atau menyediakan cara sistematis untuk menghasilkan perbaikan terhadap masalah kebergunaan, namun cukup mudah untuk memperbaiki masalah tersebut karena masalah yang ditemukan memiliki referensi prinsip-prinsip kebergunaan yang dilanggar. Dalam hal *severity ratings*, metode pengujian *heuristic evaluation* menggunakan pendekatan *single scale* sebagai berikut [21]:

0 = Tidak ada permasalahan kebergunaan

1 = Hanya *cosmetic problem*, sehingga tidak perlu diperbaiki kecuali masih tersedia waktu untuk pengembangan projek

2 = Minor usability problem, perbaikan ini bersifat prioritas rendah.

3 = Major usability problem, penting untuk memperbaiki permasalahan ini dan menjadi prioritas tinggi.

4 = *usability catastrophe*, wajib atau harus diperbaiki sebelum produk dirilis.

Dalam proses pengujian menggunakan *heuristic evaluation*, terdapat 10 prinsip yang menjadi acuan pengujian, prinsip tersebut umumnya dikenal dengan sebutan “*Ten usability heuristic*”. Beberapa prinsip tersebut adalah [22]:

1. *Visibility of system status*
2. *Match between system and the real world*
3. *User control and freedom*
4. *Consistency and standards*
5. *Error prevention*
6. *Recognition rather than recall*
7. *Flexibility and efficiency of use*
8. *Aesthetic and minimalist design*
9. *Help user recognize, diagnose, and recover from errors*
10. *Help and documentation*

2.2 Penelitian Terdahulu

Dalam proses pelaksanaan penelitian ini, dilakukan suatu teknik pengumpulan data yaitu studi literatur terhadap beberapa penelitian terdahulu yang relevan dengan topik penelitian ini. Hal tersebut dilakukan untuk mendukung pelaksanaan penelitian ini dan menghindari plagiarisme. Pada bagian ini akan dijabarkan beberapa penelitian relevan yang terdahulu.

Terdapat penelitian yang dilakukan oleh Ridhoni dkk., pada tahun 2022 yang berjudul *Pengembangan Aplikasi Alquran Online dengan Memanfaatkan REST API*. Metode pengembangan perangkat lunak yang digunakan dalam penelitian ini adalah metode *waterfall*. Penelitian ini memanfaatkan REST API sebagai sumber data yang akan ditampilkan pada halaman pengguna sesuai dengan hasil desain yang telah dibuat sebelumnya. Hasil dari penelitian ini berupa produk Al-Qur'an digital yang dapat menjadi salah satu alternatif media untuk melantunkan ayat-ayat suci. Salah satu kesimpulan penelitian ini adalah bahwa REST API dapat dimanfaatkan untuk mengembangkan aplikasi sesuai dengan kebutuhan pengguna [23].

Penelitian selanjutnya dilakukan oleh Fikri & Arthawan pada tahun 2020 yang berjudul *Rancang Bangun Sistem Informasi Buku Tamu pada Dinas Pemuda, Olahraga, dan Pariwisata Kota Balikpapan dengan Metode Personal Extreme Programming*. Penelitian ini menghasilkan produk berupa sistem informasi buku tamu yang selanjutnya digunakan pada dispora Kota Balikpapan. Adapun metode pengembangan perangkat lunak yang digunakan dalam penelitian ini adalah *Personal Extreme Programming (PXP)*. Metode PXP yang digunakan dalam penelitian ini telah mempermudah proses pengembangan perangkat lunak yang dilakukan secara individual. Disamping itu, fungsionalitas sistem yang dikembangkan juga berjalan sebagaimana mestinya. Hal ini dibuktikan dengan pengujian menggunakan metode *Black Box Testing* yang diterapkan pada sistem tersebut. Salah satu kesimpulan dari penelitian ini adalah bahwa penggunaan PXP dalam pengembangan perangkat lunak memberikan dampak baik pada produk yang dikembangkan, sehingga penggunaan metode PXP pada pengembangan sistem tersebut merupakan pilihan yang tepat untuk pemrograman secara individual [24].

Penelitian lainnya dilakukan oleh Rambe dkk., pada tahun 2020 dengan judul *UML Modelling and Black Box Testing Methods in the School Payment Information System*. Penelitian ini menggunakan metode *Unified Modelling Language* (UML) dalam pengembangan perangkat lunak. Disamping itu, penelitian ini juga menggunakan metode *black box testing* dalam pengujian perangkat lunak yang telah dikembangkan. Penelitian ini menghasilkan suatu produk berupa sistem informasi pembayaran sekolah yang mempermudah staff TU dalam melakukan pendataan transaksi pembayaran sekolah. Salah satu kesimpulan dari penelitian ini adalah bahwa *black box testing* dalam pengujian perangkat lunak memberikan fokus utama pada fungsionalitas aplikasi. Sehingga pengguna dapat melakukan pengujian secara langsung tanpa harus memahami proses internal perangkat lunak tersebut [25].

Kemudian penelitian yang dilakukan oleh Leonardo dkk., pada tahun 2020 berjudul *Pemanfaatan Teknologi Firebase dalam Pengembangan Aplikasi Pengelolaan Stok Barang Berbasis Mobile pada Rumah Makan Nakamse Malang*. Penelitian ini menggunakan metode pengembangan *waterfall* dengan memanfaatkan *Unified Modeling Language* (UML). Dalam proses pengembangan aplikasi pada penelitian tersebut, peneliti menggunakan *firebase* untuk menjadi tempat atau basis penyimpanan data. *Firestore* sendiri menyediakan berbagai library yang dapat digunakan untuk mempermudah dan mempercepat pengembangan suatu aplikasi. Penelitian ini menghasilkan suatu produk berupa aplikasi *mobile* yang telah diuji dengan menggunakan metode *black box testing* dan juga pengujian *usability*. Adapun skor yang didapat yaitu *acceptable* dan *good*. Berdasarkan hal tersebut, maka dapat disimpulkan bahwa pemanfaatan *firebase* dalam pengembangan suatu aplikasi dapat memberikan nilai positif bagi aplikasi yang dikembangkan [26].

Penelitian terkait *React JS* dilakukan oleh Huda & Fernando pada tahun 2021 berjudul *E-Ticketing Penjualan Tiket Event Musik di Wilayah Lampung pada Karcismu Menggunakan Library ReactJS*. Penelitian ini menghasilkan suatu perangkat lunak yang dikembangkan dengan memanfaatkan Pustaka *javascript* bernama *React JS*. *React JS* merupakan Pustaka *javascript* yang dapat digunakan untuk membangun tampilan pengguna pada aplikasi web dengan menggunakan

konsep pengembangan berbasis komponen. Metode pengembangan perangkat lunak yang digunakan yaitu *GRAPPLE (Guidelines for Rappid Application Engineering)*. Adapun metode pengujian pada perangkat lunak ini menggunakan *black box testing*. Hasil pengujian memberikan hasil sebesar 100% sehingga dapat disimpulkan bahwa perangkat lunak yang dikembangkan dengan memanfaatkan *React JS* memiliki kriteria sangat baik [27].

Beberapa penelitian relevan yang dijabarkan sebelumnya membantu proses pelaksanaan penelitian ini, khususnya dalam menentukan posisi penelitian. Metode pengembangan, pengujian, dan *web service* perangkat lunak yang digunakan dapat berbeda-beda sesuai dengan kebutuhan. Pengembangan Al-Qur'an digital berbasis aplikasi web ini menggunakan metode *Personal Extreme Programming (PXP)* dan pengujian aplikasinya menggunakan *black box testing*. Penelitian ini juga memanfaatkan REST API sebagai *web service* yang digunakan untuk menampilkan berbagai data yang dibutuhkan ke halaman pengguna. Salah satu pustaka *javascript* yang biasa digunakan dalam pengembangan web modern, yaitu *React JS*, juga menjadi salah satu pondasi pengembangan aplikasi ini. Selanjutnya layanan *firebase* yang disediakan oleh google juga menjadi media penyimpanan data dan autentikasi pengguna pada perangkat lunak Al-Qur'an digital ini.

2.3 Analisis Kompetitor

Pada penelitian ini dilakukan analisis terhadap aplikasi sejenis yang beredar di internet, baik dari sisi kelebihan maupun kelemahan. Adapapun aplikasi yang akan dianalisis yaitu aplikasi Muslim Pro, dengan alasan bahwa berdasarkan riset menggunakan metode kuisisioner yang telah dilakukan sebelumnya, aplikasi Muslim Pro memiliki jumlah pengguna terbanyak, yaitu 33 pengguna dari total 60 responden.

Aplikasi ini dianalisis dari berbagai aspek, mulai dari kelebihan, kekurangan, ulasan pengguna, dan fleksibilitas. Tujuan dari pelaksanaan analisis kompetitor ini adalah untuk menjadi acuan dan referensi dalam proses pengembangan perangkat lunak Al-Quran digital pada penelitian ini. Dengan begitu diharapkan aplikasi Al-Qur'an digital pada penelitian ini memiliki unsur keterbaruan dan memenuhi kekurangan yang ada pada aplikasi Muslim Pro sebagai kompetitor. Tentunya keterbaruan aplikasi yang dikembangkan pada penelitian ini sesuai dengan tujuan penelitian dan tidak meluas ke berbagai aspek diluar lingkup Al-Qur'an digital.

Aplikasi Muslim Pro sendiri tentunya memiliki berbagai kelebihan mulai dari kelengkapan fitur, pelayanan, hingga jumlah pengguna yang begitu banyak. Namun jika dilihat dari sisi lainnya, aplikasi ini masih memiliki kekurangan. Di antaranya adalah iklan yang begitu banyak sehingga menyulitkan pengguna yang ingin fokus saat membaca Al-Qur'an atau melakukan kegiatan lainnya dalam aplikasi. Beberapa fitur juga memiliki kekurangan jika dilihat dari segi pengalaman pengguna, masih ada fitur-fitur yang sifatnya tidak sesuai dengan harapan pengguna dan terkadang menyulitkan, seperti fitur pencarian. Di samping itu masih ada fitur yang belum disediakan, seperti pencatatan riwayat hafalan Al-Qur'an dan *murojaah*. Jika dilihat dari segi pengguna, Muslim Pro sendiri memiliki jumlah pengguna yang begitu banyak, hal ini tidak hanya didasarkan pada hasil kuisisioner yang telah dilakukan sebelumnya, namun juga berdasarkan informasi yang ditampilkan pada halaman *playstore*, yaitu Muslim Pro memiliki jumlah pengguna sebanyak 100 juta unduhan. Namun dari banyaknya jumlah pengguna tersebut, tidak sedikit yang memberikan ulasan negatif karena kekecewaan terhadap aplikasi Muslim Pro itu sendiri.

2.4 Tools yang Digunakan

2.4.1 Visual Studio Code

Visual Studio Code (VS Code) merupakan suatu aplikasi pengembangan perangkat lunak yang didesain untuk mempermudah pengembangan aplikasi web, *mobile*, dan *cloud* dengan menggunakan berbagai bahasa yang tersedia untuk mendukung SDLC. Disamping itu, VS Code memiliki *debugger* bawaan dan telah terintegrasi dengan dukungan untuk mesin control versi GIT yang populer. Penggunaan VS Code sebagai teks editor, dapat digunakan melalui fail atau folder berisi beberapa projek fail. Dalam hal instalasi, VS Code dapat dilakukan instalasi dan dijalankan pada sistem operasi windows, macOS, dan Linux. Kelebihan lainnya yang terdapat pada VS Code adalah adanya fitur *extensions* yang dapat mempermudah pengguna dalam proses mengembangkan aplikasi [28].

2.4.2 Java Script

Java script (JS) merupakan salah satu bahasa pemrograman yang digunakan dalam pengembangan web. Pada umumnya, JS digunakan untuk membuat halaman web lebih interaktif dan dinamis. Sehingga JS sering kali ditemukan pada hampir seluruh web modern. Di samping itu, JS juga dapat digunakan dalam pengembangan program dari sisi *backend* dengan menggunakan node JS, serta pengembangan aplikasi desktop. Pada pengimplementasian java script dalam pengembangan web, pengguna dapat menyisipkan tag `<script>` pada halaman html. Dengan begitu, halaman html dapat terkoneksi dengan kode program java script yang akan dikembangkan [29].

2.4.3 React JS

React merupakan salah satu *library* java script yang dapat digunakan untuk membuat tampilan pengguna lebih interaktif dengan menggunakan konsep komponen. React dikembangkan oleh facebook pada tahun 2011, hingga pada tahun 2013, versi pertama react dirilis secara publik dan dapat digunakan secara gratis. Pada saat ini, *library* tersebut telah digunakan oleh berbagai aplikasi web dan *mobile*, termasuk di antaranya adalah facebook, Instagram, netlix, Airbnb, dan lain-

lain. Facebook sendiri merancang React sebagai solusi atas kebutuhannya dalam pengembangan web. React juga dapat didefinisikan sebagai *library* yang dapat membuat dan menyusun komponen untuk membangun tampilan antarmuka pengguna. Komponen react sendiri merupakan bagian independen yang sifatnya *reusable* atau dapat digunakan kembali, sehingga dapat meningkatkan efektifitas dalam pengembangan dan pemeliharaan kode program [30].

2.4.4 Application Programming Interface (API)

Application Programming Interface (API) merupakan salah satu bagian terbesar dalam pengembangan web. Pada tahun 2013, terdapat lebih dari 10.000 API yang telah dipublikasi oleh berbagai perusahaan untuk digunakan secara terbuka. API dapat didefinisikan sebagai alat yang membuat sekumpulan data dapat dicerna dengan mudah oleh komputer. Contoh pengimplementasi atau pemanfaatan API adalah ketika pengguna menghadapi tugas yang sangat menghabiskan banyak waktu, misalnya menyalin info kontak untuk seribu pelanggan dari satu situs ke situs lain. Jika dilakukan secara manual, tentu itu sangat melelahkan dan menghabiskan waktu. Sehingga API diciptakan untuk menjadi solusi atas permasalahan tersebut, yaitu dengan menjadi alat untuk membuat data lebih mudah dikerjakan. API juga dapat didefinisikan sebagai program yang berjalan pada server yang menangani lalu lintas web. Ketika suatu situs mengambil data dari situs lain, maka situs yang menyediakan data bertindak sebagai server, situs yang mengambil data bertindak sebagai klien, sehingga server dan klien terhubung melalui API [31].

2.4.5 Firebase

Firebase merupakan salah satu layanan google yang dirilis pada tahun 2016. Firebase dikembangkan sebagai penyedia infrastruktur yang dapat digunakan untuk pengembangan aplikasi web, android, dan IOS. Beberapa layanan atau fitur yang disediakan oleh firebase di antaranya adalah *authentication* untuk membuat autentikasi pengguna, *realtime database & firestore* untuk manajemen basis data, *firebase hosting* untuk menjadi media *hosting* aplikasi yang dikembangkan, *firebase test lab* dapat digunakan untuk pengujian aplikasi, *firebase crash reporting*

sebagai media laporan apabila ada kesalahan dalam sistem, *firebase cloud messaging* untuk mempermudah menyampaikan pesan tanpa harus menghabiskan biaya, *firebase remote config* untuk konfigurasi sistem, dan berbagai fitur firebase lainnya. Dengan adanya berbagai fitur tersebut, hal ini tentu mempermudah pengguna dalam proses pengembangan aplikasi, sehingga pengguna tidak perlu memikirkan atau membuat sendiri *backend* yang dibutuhkan [32].

III. METODOLOGI PENELITIAN

3.1 Waktu dan Tempat

3.1.1 Waktu

Tabel 6. Waktu pelaksanaan penelitian

NO	Kegiatan Penelitian	Jul	Aug	Sept	Oct	Nov	Des	Jan
1	<i>Requirements</i>							
2	<i>Planning</i>							
3	<i>Iteration initialization</i>							
4	<i>Design</i>							
5	<i>Implementation</i>							
6	<i>System Testing</i>							
7	<i>Retrospective</i>							
8	Penulisan karya ilmiah							
9	Seminar Hasil							
10	Ujian Komprehensif							

Tabel 6 merupakan waktu pelaksanaan penelitian mulai dari tahapan *requirements* hingga ujian komprehensif. Secara umum, penelitian ini dilaksanakan selama 7 bulan mulai dari bulan Juli 2023 hingga Januari 2024.

3.1.2 Tempat

Penelitian ini dilaksanakan di lingkungan laboratorium teknik komputer Universitas Lampung, Jl. Prof. Dr. Ir. Sumantri Brojonegoro No.1, Gedong Meneng, Kec. Rajabasa, Kota Bandar Lampung, Lampung.

3.2 Alat dan Bahan Penelitian

3.2.1 Alat Penelitian

Pada penelitian ini, digunakan beberapa alat yang digunakan untuk mendukung dan menunjang setiap tahapan yang sesuai dengan prosedur penelitian. Adapun alat pada penelitian ini terbagi atas dua jenis, yaitu perangkat keras dan perangkat lunak.

Tabel 7 dan tabel 8 di bawah ini merupakan tabel daftar alat-alat yang digunakan dalam penelitian ini:

Tabel 7. Perangkat keras

Perangkat Keras	Fungsi
Komputer Intel Core 5 gen 12	Membuat desain dan implementasi kode program
Laptop Intel Core 5 gen 10	Membuat desain dan implementasi kode program
Mouse	Menggerakkan kursor
Keyboard	Mengetik kata atau kalimat

Tabel 8. Perangkat lunak

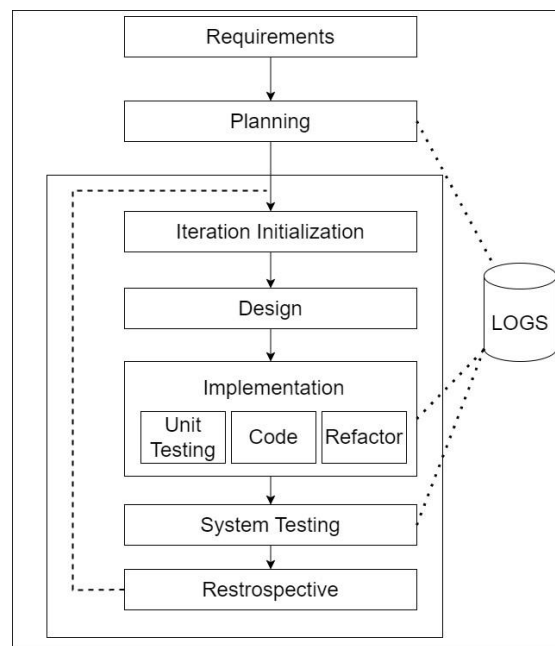
Perangkat Lunak	Fungsi
Visual Studio Code	Kode editor untuk mengolah kode program
Firebase	Menjadi penyedia layanan <i>back end</i>
Google chrome (browser)	Media implementasi web
React JS	<i>Library</i> pengembangan kode program
API E-Quran.id	Media integrasi data Al-Qur'an
HTML dan CSS	Kode program untuk pengembangan web
Javascript	Bahasa pemrograman pengembangan web
Figma	<i>Tools</i> untuk membuat desain
Diagrams.net atau Draw.io	<i>Tools</i> untuk membuat diagram

3.2.2 Bahan Penelitian

Pada penelitian ini, digunakan beberapa bahan penelitian untuk mendukung informasi dan proses pengembangan aplikasi Al-Qur'an digital ini.

1. Data pra survei yang didapat dengan metode kuisisioner.
2. Beberapa penelitian terdahulu yang relevan.
3. Analisis kompetitor berupa perangkat lunak yang memiliki kesamaan fitur dalam beberapa hal.
4. Jurnal nasional dan internasional yang digunakan sebagai acuan penelitian.
5. API Equran.id yang menjadi sumber data Al-Qur'an dalam pengembangan sistem pada penelitian ini.

3.3 Tahapan Penelitian



Gambar 4. Tahapan PXP [3]

Gambar 4 merupakan berbagai tahapan metode *personal extreme programming* yang akan dilaksanakan dalam penelitian ini. Tahapan tersebut secara umum terdiri atas *requirements*, *planning*, *iteration initialization*, *design*, *implementation*, *system testing*, dan *retrospective*.

3.3.1 Requirements

Membuat daftar kebutuhan fungsional dan non-fungsional terkait dengan sistem yang akan dikembangkan. Kebutuhan tersebut disesuaikan dengan kebutuhan pengguna yang diambil dari hasil pengumpulan data. Dalam hal pengumpulan data, metode yang digunakan yaitu studi literatur, kuisisioner, dan observasi.

Penggunaan metode studi literatur penelitian ini adalah untuk mengumpulkan informasi terkait yang kredibilitasnya teruji guna menjadi bahan referensi dan mendukung proses pelaksanaan penelitian ini. Dalam pelaksanaannya, penelitian ini memanfaatkan studi literatur untuk peninjauan pustaka, metodologi, posisi penelitian atau penelitian yang relevan, pengembangan hipotesis awal, pemilihan teknologi yang digunakan, dan riset teoritis. Proses

pengumpulan literatur ini memanfaatkan teknologi internet, salah satunya adalah *google scholar*.

Selanjutnya penelitian ini juga menggunakan metode kuisisioner dalam mengumpulkan data pra survei untuk memperkuat informasi yang telah dicantumkan pada laporan skripsi ini. Adapun metode kuisisioner yang digunakan telah disajikan dalam bentuk beberapa pertanyaan yang selanjutnya dimanfaatkan untuk memperkuat dasar alasan pemilihan topik penelitian ini. Responden yang terlibat dalam kuisisioner pada penelitian ini yaitu mahasiswa Universitas Lampung yang berjumlah 60 orang.

Teknik observasi juga dilakukan dalam proses pengumpulan data. Pada penelitian ini, teknik pengumpulan data berupa observasi digunakan untuk melakukan analisis terhadap kompetitor. Kompetitor dalam penelitian ini adalah perangkat lunak sejenis, yaitu Muslim Pro. Alasan pemilihan aplikasi ini sebagai kompetitor adalah karena aplikasi ini memiliki jumlah pengguna terbanyak dibandingkan perangkat lunak sejenis lainnya berdasarkan kuisisioner yang telah dilakukan sebelumnya. Aplikasi tersebut diobservasi dari berbagai aspek, di antaranya adalah kelebihan, kekurangan, ulasan, fleksibilitas, dan analisis SWOT (*Strong, Weakness, Opportunity, Threats*). Tujuan penggunaan analisis SWOT dalam analisis kompetitor adalah untuk membantu memahami posisi keunggulan dan kelemahan kompetitor serta peluang dan ancaman.

3.3.2 Planning

Melakukan penyusunan terhadap serangkaian tugas yang akan dikerjakan berdasarkan hasil tahapan *requirements* yang telah dibuat sebelumnya. Tahapan ini dilaksanakan dengan cara membagi berbagai *user stories* yang telah dibuat sebelumnya menjadi beberapa bagian sesuai dengan banyaknya jumlah iterasi yang akan dilakukan. Di dalamnya juga terdapat penggunaan *story point* untuk menentukan perkiraan waktu yang dibutuhkan dalam pengembangan setiap fitur yang ada. Dalam proses pembuatan dokumen perencanaan, dilakukan analisis fitur dengan menggunakan metode *moscow prioritization* untuk menentukan tingkat prioritas fitur yang akan dikembangkan.

3.3.3 *Iteration Initialization*

Membuat *Unified Modeling Language* (UML) untuk memperjelas fungsional perangkat lunak yang akan dikembangkan. Dalam hal ini digunakan diagram *use case* untuk memberikan penjelasan hubungan antara aktor atau pengguna dengan sistem. Pembuatan UML ini tentunya disesuaikan dengan dokumen *requirements* yang telah dibuat pada tahap sebelumnya. Tujuannya adalah untuk memberikan kemudahan bagi para pembaca dalam memahami bagaimana sistem ini bekerja dan berinteraksi dengan pengguna. Di samping itu, pembuatan diagram *use case* ini adalah untuk memberikan kemudahan dalam proses pengembangan perangkat lunak Al-Quran digital pada penelitian ini.

3.3.4 *Design*

Membuat desain hubungan antar kelas dengan menggunakan teknik permodelan *class diagram*. Disamping itu, pada tahapan ini juga membuat desain tampilan pengguna secara kasar atau kualitas yang rendah, tujuan dari pembuatan desain dengan kualitas yang rendah (*low fidelity*) salah satunya adalah untuk meminimalisir waktu yang akan digunakan dalam proses pengembangan sistem menggunakan metode PXP. Kedua desain tersebut diterapkan untuk mempermudah proses pengembangan program dan membantu pembaca dalam memahami konsep program yang dikembangkan.

3.3.5 *Implementation*

Melakukan implementasi pengembangan perangkat lunak dengan memanfaatkan desain yang telah dibuat sebelumnya. Pada tahap ini dilakukan pengembangan kode program, penyederhanaan kode program, dan pengujian fitur yang telah dibuat. Dalam hal pengembangan kode program, digunakan beberapa *tools* yang dapat mempermudah dalam mengembangkan aplikasi Al-Qur'an digital. *Tools* tersebut di antaranya adalah *Visual Studio Code*, *React JS Library*, *Firebase*, dan beberapa bahasa penanda seperti *HTML & CSS*, serta bahasa pemrograman *java script*. Disamping itu, proses implementasi kode program juga memanfaatkan REST API equran.id. API ini dikembangkan oleh komunitas Muslim Dev dengan menggunakan sumber data dari Aplikasi Quran Kementerian Agama Republik

Indonesia. Adapun data yang ditampilkan pada API tersebut di antaranya adalah data-data fundamental yang berhubungan dengan Al-Qur'an, seperti data surat, ayat, terjemahan, tafsir, dan berbagai data lainnya. Penyederhanaan kode dilakukan dengan menekankan pada salah satu prinsip dasar pengembangan perangkat lunak, yaitu DRY yang berarti *Don't Repeat Yourself*. Adapun pengujian unit fitur pada penelitian ini menggunakan metode *black box testing*.

3.3.6 *System Testing*

Melakukan pengujian sistem terhadap seluruh fitur yang telah dikembangkan sejak awal (mulai dari iterasi ke-1). Tujuannya adalah untuk memastikan setiap fitur dapat berjalan sesuai dengan fungsinya. Pengujian ini menggunakan metodologi *User Acceptance Test (UAT)* dengan memanfaatkan metode *alpha testing* menggunakan *black box*. Metode ini melakukan pengujian terhadap fungsionalitas sistem agar sesuai dengan kebutuhan pengguna. Alasan penggunaan metode ini adalah untuk meminimalisir waktu yang digunakan tanpa meninggalkan perhatian khusus terhadap fungsional aplikasi.

3.3.7 *Retrospective*

Melakukan evaluasi terhadap kesalahan-kesalahan sebelumnya agar tidak terjadi pengulangan kesalahan yang sama untuk iterasi selanjutnya. Pada tahap ini juga dilakukan proses pengujian kebergunaan atau *usability testing* dengan menggunakan metode *System Usability Scale (SUS)* dan *Heuristic Evaluation (HE)* untuk menguji nilai *usability* aplikasi dari sudut pandang pengguna dalam menggunakan aplikasi Al-Qur'an digital. Dalam pengujian SUS, nilai yang diuji adalah nilai *acceptable* aplikasi oleh pengguna akhir (*end user*). Adapun jumlah responden yang terlibat dalam pengujian menggunakan metode SUS adalah sebanyak 13 orang, 10 orang merupakan mahasiswa Universitas Lampung dan 3 lainnya merupakan orang yang pernah atau sedang menghafal Al-Qur'an. Hal ini berdasarkan penelitian relevan yang terdahulu, bahwa responden berjumlah 8 hingga 13 orang dalam pengujian SUS akan menghasilkan penilaian yang efektif [5]. Selanjutnya terkait dengan pengujian menggunakan metode *heuristic evaluation*, nilai yang diuji adalah nilai *severity rating*. Adapun jumlah evaluator

yang terlibat sebagai penguji dalam metode ini adalah 3 orang yang memiliki pengalaman dibidang pengembangan web. Penentuan jumlah responden tersebut disebabkan karena penelitian sebelumnya menyebutkan bahwa tiga hingga lima evaluator merupakan rekomendasi terbaik pada pengujian *heuristic evaluation* [6].

V. PENUTUP

5.1 Kesimpulan

Berdasarkan penelitian yang telah dilakukan, maka dapat disimpulkan bahwa:

1. Aplikasi Al-Qur'an digital pada penelitian ini berhasil dikembangkan dengan menggunakan metode *Personal Extreme Programming* (PXP) dalam kurun waktu 7 iterasi atau 53 hari. Realisasi waktu ini lebih cepat daripada estimasi waktu yang telah direncanakan pada tahap *planning*, dengan estimasi waktu 60 hari. Namun iterasi yang dikerjakan lebih banyak daripada estimasi jumlah iterasi pada tahap *planning*, dengan estimasi iterasi sebanyak 6 iterasi.
2. Berdasarkan hasil pengujian menggunakan metode *black box testing*, maka didapatkan kesimpulan bahwasanya aplikasi yang dikembangkan sudah memiliki fungsionalitas yang baik dan sesuai dengan kebutuhan pengguna. Adapun fitur yang diuji berjumlah 55 fitur dengan total keseluruhan pengujian dapat diterima/berhasil.
3. Berdasarkan hasil pengujian menggunakan metode *System Usability Scale* (SUS) oleh 13 responden, maka didapatkan skor akhir dengan nilai 88,46. Hal ini menandakan bahwa aplikasi yang dikembangkan sudah memenuhi nilai kebergunaan atau *usability* yang dapat diterima (*acceptable*).
4. Berdasarkan hasil pengujian menggunakan metode *Heuristic Evaluation* (HE) oleh 3 evaluator, maka didapatkan skor akhir *severity rating* dengan nilai 0,03 yang apabila dilakukan pembulatan maka menjadi 0. Sehingga hasil tersebut menunjukkan bahwa aplikasi yang dikembangkan tidak memiliki masalah *usability* atau kebergunaan.

5.2 Saran

Adapun saran untuk penelitian selanjutnya adalah sebagai berikut.

1. Pengujian fungsionalitas aplikasi menggunakan *black box testing*, selanjutnya perlu dilakukan pengujian menggunakan *white box testing* guna meningkatkan hasil pengujian.
2. *Backend* pada aplikasi yang dikembangkan menggunakan *firebase* (layanan google), perlu *backend* kustom untuk meningkatkan fleksibilitas dari sisi *backend* pada aplikasi.
3. Perlu adanya peningkatan kualitas fitur murojaah, salah satunya dapat memanfaatkan teknologi *Artificial Intelligence* (AI) sehingga sistem dapat mengoreksi secara otomatis apabila ada kesalahan baca.

DAFTAR PUSTAKA

- [1] A. G. Pereira, T. M. Lima and F. C. Santos, "Industry 4.0 and Society 5.0: Opportunities and Threats," *International Journal of Recent Technology and Engineering (IJRTE)*, vol. 8, no. 5, pp. 3305-3308, 2020.
- [2] M. F. Mubarak and M. F. Romdhoni, "Digitalisasi al-Qur'an dan Tafsir Media Sosial di Indonesia," *Jurnal Iman dan Spiritualitas*, vol. 1, no. 1, pp. 110-114, 2021.
- [3] Y. Dzhurov, I. Krasteva and S. Illieva, "Personal Extreme Programming – An Agile Process for Autonomous Developers," in *Proceedings of International Conference on Software, Services & Semantic Technologies*, Bulgaria, 2009.
- [4] A. Verma, A. Khatana and S. Chaudhary, "A Comparative Study of Black Box Testing and White Box Testing," *International Journal of Computer Science and Engineering*, vol. 5, no. 12, pp. 301-304, 2017.
- [5] T. S. Tullis and J. N. Stetson, "A Comparison of Questionnaires for Assessing Website Usability," in *Proceedings of UPA 2004 Conference*, Minneapolis, 2004.
- [6] J. Nielsen and R. Molich, "Heuristic Evaluation of User Interface," in *CHI Conference*, Washington, 1990.
- [7] S. A. Asri, I. G. A. M. Sunaya, E. Rudiastari and W. Setiawan, "Web Based Information System for Job Training Activities Using Personal Extreme Programming (PXP)," in *The 2nd International Joint Conference on Science and Technology (IJCST)*, Nusa Dua Bali, 2018.
- [8] R. Kumar, *Research Methodology: A Step-By-Step Guide for Beginners 3rd Edition*, London: Sage Publication, 2011.

- [9] Ni'matuzahroh and S. Prasetyaningrum, *Observasi: Teori dan Aplikasi dalam Psikologi*, Malang: Universitas Muhammadiyah Malang, 2018.
- [10] I. Sommerville, *Software Engineer Ninth Edition*, Boston: Addison-Wesley, 2011.
- [11] Y. Wautelet, M. Kolp and I. Mirbel, "Unifying and Extending User Story Models," in *International Conference on Advanced Information Systems Engineering (CAiSE)*, Thessaloniki, 2014.
- [12] Monochristou, V and M. Vlachopoulou, "Requirements Specification using User Stories," in *Agile Software Development Quality Assurance*, United States, Idea Group Inc, 2007, pp. 71-89.
- [13] M. Seidl, M. Scholz, C. Huemer and G. Kappel, *UML @ Classroom An Introduction to Object-Oriented Modeling*, Switzerland: Springer, 2015.
- [14] A. B. Chaudhuri, *Flowchart and Algorithm Basics: The Art of Programming*, Mercury Learning and Information, 2020.
- [15] I. Afrianto, A. Heryandi, A. Finadhita and S. Atin, "User Acceptance Test For Digital Signature Application In Academic Domain To Support The Covid-19 Work From Home Program," *International Journal of Information System & Technology*, vol. 5, no. 3, pp. 270-280, 2021.
- [16] M. E. Khan, "Different Approaches to Black Box Testing Technique for Finding Errors," *International Journal of Software Engineering & Applications*, vol. 2, no. 4, pp. 31-40, 2011.
- [17] J. Brooke, "SUS: A 'Quickly and Dirty' Usability Scale," in *Usability Evaluation in Industry*, United Kingdom, Taylor & Francis, 1996, pp. 189-194.
- [18] U. Ependi, T. B. Kurniawan and F. Panjaitan, "System Usability Scale VS Heuristic Evaluation: A Review," *Jurnal SIMETRIS*, vol. 10, no. 1, pp. 65-74, 2019.

- [19] A. Bangor, P. Kortum and J. Miller, "Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale," *Journal of Usability Studies*, vol. 4, no. 3, pp. 114-123, 2009.
- [20] J. Nielsen, "Finding Usability Problems Through Heuristic Evaluation," in *CHI Conference*, Washington, 1992.
- [21] J. Nielsen, *Usability Engineering*, San Fransisco: Academic Press, Inc., 1994.
- [22] J. Nielsen, "Enhancing the Explanatory Power of Usability Heuristics," in *CHI Conferen ce*, Boston, 1994.
- [23] I. W. Ridhoni, M. Sholeh and U. J. Basuki, "Pengembangan Aplikasi Alquran Online dengan Memanfaatkan REST API," *Jurnal Device*, vol. 12, no. 2, pp. 1-7, 2022.
- [24] A. M. Fikri and I. P. D. Arthawan, "Rancang Bangun Sistem Informasi Buku Tamu pada Dinas Pemuda, Olahraga, dan Pariwisata Kota Balikpapan dengan Metode Personal Extreme Programming," *Multitek Indonesia: Jurnal Ilmiah*, vol. 14, no. 2, pp. 101-109, 2020.
- [25] B. H. Rambe, R. Pane, D. Irmayani, M. Nasution and I. R. Munthe, "UML Modeling and Black Box Testing Methods in the School Payment," *Jurnal Mantik*, vol. 4, no. 3, pp. 1634-1640, 2020.
- [26] R. Leonardo, I. Arwani and D. E. Ratnawati, "Pemanfaatan Teknologi Firebase dalam Pengembangan Aplikasi Pengelolaan Stok Barang Berbasis Mobile pada Rumah Makan Nakamse Malang," *Jurnal JUST-SI*, vol. 1, no. 1, pp. 1-11, 2020.
- [27] A. M. S. Huda and Y. Fernando, "E-Ticketing Penjualan Tiket Event Musik di Wilayah Lampung pada Karcismu Menggunakan Library ReactJS," *Jurnal Teknologi dan Sistem Informasi (JTSI)*, vol. 2, no. 1, pp. 96-103, 2021.
- [28] A. D. Sole, *Visual Studio Code Distilled: Evolved Code Editing for Windows, macOS, and Linux - Second Edition*, Cremona, Italy: Apress, 2021.

- [29] M. F. Adiwisastro, A. B. Hikamh and A. I. Warnilah, Dasar Pemrograman Web, Grobongan, Jawa Tengah: CV. Sarnu Untung, 2019.
- [30] C. Minnick, Beginning ReactJS Foundations: Building User Interfaces with ReactJS - An Approachable Guide, Canada: Jhon Wiley & Sons, Inc, 2022.
- [31] B. Cooksey, An Introduction to APIs, Zapier, Inc, 2014.
- [32] L. Moroney, The Definitive Guide to Firebase, Washington, USA: APRESS, 2017.