

**EVALUASI DAN PENGEMBANGAN FITUR PADA WEBSITE QUISION:
PLATFORM PENGHUBUNG PENELITI DENGAN RESPONDEN
MENGUNAKAN METODE *AGILE DEVELOPMENT***

(Skripsi)

Oleh

**AFRIZAL YOGI PRATAMA
NPM 2115061032**



**FAKULTAS TEKNIK
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2025**

**EVALUASI DAN PENGEMBANGAN FITUR PADA WEBSITE QUISION:
PLATFORM PENGHUBUNG PENELITI DENGAN RESPONDEN
MENGUNAKAN METODE *AGILE DEVELOPMENT***

Oleh

AFRIZAL YOGI PRATAMA

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
2025**

ABSTRAK

EVALUASI DAN PENGEMBANGAN FITUR PADA WEBSITE QUISION: PLATFORM PENGHUBUNG PENELITI DENGAN RESPONDEN MENGUNAKAN METODE *AGILE DEVELOPMENT*

Oleh

AFRIZAL YOGI PRATAMA

Penelitian ini mengembangkan platform Quision berbasis website yang menghubungkan peneliti dengan responden untuk pengumpulan data kuesioner. Menggunakan metode *Agile Development* dengan kerangka kerja *Scrum*, penelitian dilakukan dalam empat iterasi yang mencakup 14 *user story*. Fokus pengembangan meliputi penyempurnaan fitur transparansi data responden, diversifikasi produk (responden hemat, umum, dan wawancara), sistem pencarian dan filter kuesioner, serta optimasi proses konfirmasi dan pencairan insentif. Hasil *usability testing* menunjukkan bahwa platform telah memenuhi kebutuhan fungsional pengguna dengan *success rate* 100%, *time-based efficiency* 0,0264 *goals*/detik, dan *error rate* 26,7%. Angka ini mengindikasikan bahwa fitur yang dikembangkan berhasil mencapai tujuan dalam hal keberhasilan fungsional, namun masih memerlukan penyempurnaan antarmuka untuk mengurangi tingkat kesalahan dan meningkatkan pengalaman pengguna secara keseluruhan. Penelitian ini tidak hanya berhasil menciptakan solusi teknis yang efektif bagi permasalahan pengumpulan data penelitian, tetapi juga membuktikan kelayakan pendekatan *Agile* dalam mengembangkan platform akademik yang responsif terhadap kebutuhan pengguna.

Kata kunci : *Scrum*, Platform Kuesioner, Quision, *Usability Testing*

ABSTRACT

EVALUATION AND FEATURES DEVELOPMENT OF THE QUISION WEBSITE: A PLATFORM CONNECTING RESEARCHERS WITH RESPONDENTS USING AGILE DEVELOPMENT METHOD

By

AFRIZAL YOGI PRATAMA

This study develops Quision, a web-based platform connecting researchers with respondents for questionnaire data collection. Employing Agile Development with Scrum framework, the research was conducted through four iterations covering 14 user stories. Development focused on enhancing respondent data transparency, product diversification (basic, standard, and interview packages), questionnaire search/filter system, as well as optimization of incentive confirmation and disbursement processes. Usability testing results show the platform has met users' functional requirements with a 100% success rate, time-based efficiency of 0.0264 goals/second, and 26.7% error rate. These figures indicate that while the developed features successfully achieved their functional objectives, interface improvements are still needed to reduce error rates and enhance overall user experience. This study has not only created an effective technical solution for research data collection problems, but has also proven the viability of the Agile approach in developing academic platforms that are responsive to user needs.

Keywords: Scrum, Questionnaire Platform, Quision, Usability Testing

Judul Skripsi : **EVALUASI DAN PENGEMBANGAN FITUR
PADA WEBSITE QUISION: PLATFORM
PENGHUBUNG PENELITI DENGAN
RESPONDEN MENGGUNAKAN METODE
AGILE DEVELOPMENT**

Nama Mahasiswa : **Afrizal Yogi Pratama**

Nomor Pokok Mahasiswa : **2115061032**

Program Studi : **S1 Teknik Informatika**

Jurusan : **Teknik Elektro**

Fakultas : **Teknik**



1. Komisi Pembimbing

Pembimbing Utama

Pembimbing Pendamping


Yessi Mulyani, S.T., M.T.
NIP. 197312262000122001


Rio Ariesta Pradipta, S.Kom., M.T.I.
NIP. 198603232019031013

2. Mengetahui

Ketua Jurusan
Teknik Elektro

Ketua Program Studi
Teknik Informatika


Herlinawati, S.T., M.T.
NIP. 197103141999032001


Yessi Mulyani, S.T., M.T.
NIP. 197312262000122001

MENGESAHKAN

1. Tim Penguji

Ketua : Yessi Mulyani, S.T., M.T.



Sekretaris

: Rio Ariesta Pradipta, S.Kom., M.T.I.



Penguji

: Wahyu Eko Sulistiono, S.T., M.Sc.



2. Dekan Fakultas Teknik



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

NIP. 197509282001121002

Tanggal Lulus Ujian Skripsi : 06 Mei 2025

SURAT PERNYATAAN

Saya yang bertanda tangan dibawah ini, menyatakan dengan sebenarnya bahwa skripsi saya yang berjudul “Evaluasi dan Pengembangan Fitur Pada Website Quision: Platform Penghubung Peneliti dengan Responden Menggunakan Metode *Agile Development*” merupakan hasil karya saya sendiri, berdasarkan ide dan inovasi dari Tim PKM-K Quision, dimana saya merupakan salah satu anggota tim tersebut. Apabila pernyataan ini tidak benar dan terbukti bahwa skripsi ini merupakan salinan, maka saya bersedia menerima sanksi sesuai dengan ketentuan hukum atau akademik yang berlaku.

Bandar Lampung, 30 April 2025

Pembuat Pernyataan,



Afrizal Yogi Pratama

2115061032

RIWAYAT HIDUP



Penulis lahir di Gisting, tanggal 15 Juli 2004 dari orang tua Sriyono dan Sugiarti. Penulis telah menyelesaikan pendidikan formal di TK Al-Anwar pada tahun 2009, SDN 1 Way Laga pada tahun 2015, MTsN 2 Bandar Lampung pada tahun 2018, SMKN 1 Bandar Lampung jurusan Teknik Komputer dan Jaringan pada tahun 2021. Pada tahun 2021, penulis diterima sebagai mahasiswa Program Studi Teknik Informatika Universitas Lampung melalui jalur SBMPTN. Selama menjadi mahasiswa, penulis aktif dalam berbagai kegiatan. Penulis aktif menjadi anggota Himpunan Mahasiswa Teknik Elektro Universitas Lampung, dan terlibat dalam divisi Sosial periode 2021/2022, serta menjadi anggota Generasi Baru Indonesia (Organisasi penerima beasiswa Bank Indonesia) divisi Pendidikan pada 2023 serta divisi Kemitraan dan Kerjasama pada tahun 2024. Dalam bidang akademis, penulis juga terlibat sebagai Asisten Laboratorium Teknik Digital pada 2022 sampai 2025. Pengembangan diri penulis tidak hanya dalam lingkup kampus, pada 2023 penulis ikut dalam program Studi Independen Bersertifikat Kampus Merdeka di mitra Infinite Learning dengan fokus bidang Website Development dan berhasil mendapat nominasi *Top 5 Best Product* sebagai *Project Manager*. Pada tahun 2024, penulis menjalani program Magang Bersertifikat Kampus Merdeka di Telkom Indonesia sebagai *Front-end Developer*. Penulis aktif mengikuti pelatihan, meliputi sertifikasi BNSP untuk *Junior Web Developer*, *Junior Network Administrator*, dan pelatihan lainnya. Penulis juga aktif dalam mengikuti kompetisi yang salah satunya adalah turut serta dalam ajang Pekan Kreativitas Mahasiswa Kewirausahaan (PKM-K) dan berhasil mencapai Pekan Ilmiah Mahasiswa Nasional Ke-36 2023 (PIMNAS). Selain itu, penulis juga menerapkan ilmu yang didapat selama perkuliahan pada lingkup masyarakat dengan ikut serta dalam pelaksanaan Kuliah Kerja Nyata (KKN) di Desa Gunung Tapa Ilir, Kecamatan Gedung Meneng, Kabupaten Tulang Bawang.

MOTTO

“Helping others eventually ends up being good for me.”

Tanjiro Kamado, *Kimetsu no Yaiba* (Swordsmith Village Arc, episode 3, 12:22)

“All big things come from small beginnings.”

James Clear, *Atomic Habits* (halaman 22)

*“If you work hard enough, assert yourself, and use your mind and imagination,
you can shape the world to your desires.”*

Malcolm Gladwell, *Outliers: The Story of Success* (halaman 151)

*“Everybody should learn to program a computer because it teaches you how to
think.”*

Steve Jobs

“The best possible outcome always comes with the bad outlook in the beginning.”

Kalimasada

“Patience is a key element of success.”

Bill Gates

“Help others, stay curious, work hard.”

Afrizal Yogi Pratama

PERSEMBAHAN

Alhamdulillahirobbilalamin

Segala puji bagi Allah SWT yang telah melimpahkan rahmat dan karunia-Nya sehingga skripsi ini dapat diselesaikan. Shalawat serta salam senantiasa dilimpahkan kepada Nabi Muhammad SAW.

Kupersembahkan Skripsi Ini Kepada

Kedua orang tuaku tercinta, Bapak Sriyono dan Ibu Sugiarti, yang telah memberikan doa dan dukungan demi kesuksesan penulis. Semoga Allah senantiasa memberikan perlindungan di dunia dan akhirat, Aamiin.

Seluruh rekan-rekan seperjuangan Teknik Informatika Universitas Lampung, Infinite Learning, Telkom Indonesia, GenBI, dan PIMNAS atas motivasi, diskusi, dan kebersamaan yang menjadikan perjalanan akademik ini penuh makna.

Seluruh dosen dan civitas akademik Jurusan Teknik Elektro Universitas Lampung, yang telah membimbing dengan ilmu, kesabaran, dan dedikasi tinggi.

Almamaterku tercinta, Universitas Lampung.

SANWACANA

Puji syukur penulis panjatkan ke hadirat Allah SWT atas rahmat, taufik, dan hidayah-Nya sehingga penulis dapat menyelesaikan skripsi ini dengan judul "Evaluasi dan Pengembangan Fitur Pada Website Quision: Platform Penghubung Peneliti dengan Responden Menggunakan Metode *Agile Development*".

Skripsi ini disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik pada Program Studi Teknik Informatika, Universitas Lampung. Proses penyusunan skripsi ini merupakan perjalanan panjang yang penuh dengan tantangan, pembelajaran, dan pengalaman berharga.

Penulis menyadari bahwa penyelesaian skripsi ini tidak terlepas dari dukungan, bimbingan, dan motivasi dari berbagai pihak. Oleh karena itu, dengan tulus hati, penulis mengucapkan terima kasih kepada:

1. Allah SWT, Tuhan Yang Maha Esa, atas segala karunia dan kekuatan yang diberikan.
2. Ibu Herlinawati, S.T., M.T., selaku Ketua Jurusan Teknik Elektro Universitas Lampung.
3. Ibu Yessi Mulyani, S.T., M.T., selaku Pembimbing Utama dan Ketua Program Studi Teknik Informatika, yang telah memberikan bimbingan dan membantu kelancaran proses pengerjaan penelitian.
4. Bapak Rio Ariestia Pradipta, S.Kom., M.T.I., selaku Pembimbing Pendamping, yang telah memberikan masukan dan arahan dalam penulisan penelitian.
5. Bapak Wahyu Eko Sulistiono, S.T., M.Sc., selaku Penguji dan Pembimbing Akademik, yang telah membagikan ilmu dan pengalaman dari awal perkuliahan sampai pelaksanaan penelitian ini.

6. Kedua orang tua dan keluarga yang senantiasa memberikan dukungan, doa, dan pengorbanan tanpa batas.
7. Nabilla Ayu Zahra yang telah memberikan dukungan dan menjadi bagian yang berarti dalam perjalanan panjang selama proses studi hingga selesai.
8. Tim Quision yaitu Bapak Esa Ghanim Fadhallah, S.Pi., M.Si., Kak Raihan, Kak Hafizh, Kak Bella, dan Kak Amel, atas kolaborasi dan pengalamannya dalam pengembangan platform sampai menuju PIMNAS.
9. Rekan-rekan Telkom Indonesia dari berbagai divisi, jurusan, dan daerah di Indonesia yaitu Dany, Fadil, Rika, Ades, Balqis, Ricky, Geha, Syiam, Bagus, Vito, Azizah, dan masih banyak lagi yang telah menjadi bagian dari perjalanan penulis dengan berbagi pengalaman, canda tawa, momen tak terlupakan, serta kenangan manis di sela-sela proyek maupun di luar jam kerja, juga para mentor yaitu Bu Sendy, Mas Ivan, Mas Iqdam, Mas Janu, Kak Laila, Kak Cindy, Kak Intan, dan lainnya yang telah berbagi cerita dalam menyelesaikan tantangan teknis maupun akademik.
10. Rekan tim Infinite Learning yaitu Imam, Ami, Anindita atas perjuangan bersama meraih *Top 5 Best Product* yang memberi motivasi saat penulis menghadapi tantangan dalam penelitian ini.
11. Seluruh mahasiswa Teknik Informatika Universitas Lampung, Afif, Ridho, Muhkito, Alvin, Rafi, Milano, Adil, Nyoman, Kak Handrio, Kak Iqbal, dan masih banyak yang tidak dapat disebutkan satu per satu, atas perjalanan dan kebersamaannya selama ini.
12. Para responden penelitian dan pengguna platform Quision yang telah memberikan umpan balik.
13. Seluruh civitas akademik Jurusan Teknik Elektro Universitas Lampung atas bantuan administratif selama proses penelitian.
14. Semua pihak yang tidak dapat disebutkan satu per satu, terima kasih atas kontribusi langsung maupun tidak langsung dalam penyelesaian skripsi ini.

Penulis menyadari bahwa skripsi ini masih jauh dari sempurna. Oleh karena itu, kritik dan saran yang membangun sangat diharapkan untuk perbaikan di masa

mendatang. Semoga karya ini dapat bermanfaat bagi perkembangan ilmu pengetahuan dan teknologi, khususnya di bidang software development.

Bandar Lampung, 30 April 2025

Penulis,

A handwritten signature in black ink, consisting of stylized, overlapping loops and strokes, representing the name Afrizal Yogi Pratama.

Afrizal Yogi Pratama

DAFTAR ISI

	Halaman
DAFTAR ISI.....	v
DAFTAR TABEL.....	viii
DAFTAR GAMBAR	x
I. PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan Penelitian	3
1.4 Manfaat Penelitian	3
1.5 Batasan Masalah.....	3
1.6 Sistematika Penulisan	4
II. TINJAUAN PUSTAKA	5
2.1 Dasar Teori.....	5
2.1.1 <i>Agile Software Development</i>	5
2.1.2 <i>Scrum</i>	7
2.1.3 <i>User Stories</i>	9
2.1.4 <i>Front End</i>	10
2.1.5 <i>Back End</i>	11
2.1.6 <i>Use Case Diagram</i>	11
2.1.7 <i>Activity Diagram</i>	13
2.1.8 <i>Hosting</i>	15
2.1.9 <i>Black Box Testing</i>	15
2.1.10 <i>Usability Testing</i>	16
2.2 Pengembangan Terdahulu.....	19
2.3 Tools yang Digunakan	21

2.3.1	<i>Content Management System (CMS)</i>	21
2.3.2	<i>WordPress</i>	22
2.3.3	<i>Application Programming Interface (API)</i>	23
2.3.4	<i>MySQL</i>	23
2.3.5	<i>Draw.io</i>	24
2.3.6	<i>Figma</i>	25
2.3.7	<i>Hypertext Preprocessor (PHP)</i>	25
2.3.8	<i>Chart.js</i>	26
III.	METODOLOGI PENELITIAN	28
3.1	Waktu dan Tempat Penelitian	28
3.1.1	Waktu Penelitian	28
3.1.2	Tempat Penelitian	29
3.2	Alat dan Bahan Penelitian	29
3.2.1	Alat Penelitian	29
3.2.2	Bahan Penelitian	30
3.3	Metode Penelitian	30
3.3.1	<i>Requirements</i>	31
3.3.2	<i>Planning</i>	31
3.3.3	<i>Development</i>	32
3.3.4	<i>Testing</i>	32
3.3.5	<i>Review</i>	33
3.3.6	<i>Deployment</i>	33
3.3.7	<i>Maintenance</i>	34
IV.	PEMBAHASAN	35
4.1	<i>Requirements</i>	35
4.2	<i>Planning</i>	56
4.3	Iterasi 1	57
4.3.1	<i>Design</i>	57
4.3.2	<i>Development</i>	59
4.3.3	<i>Testing</i>	73
4.3.4	<i>Review</i>	73
4.4	Iterasi 2	74

4.4.1	<i>Design</i>	74
4.4.2	<i>Development</i>	76
4.4.3	<i>Testing</i>	89
4.4.4	<i>Review</i>	89
4.5	Iterasi 3.....	90
4.5.1	<i>Design</i>	90
4.5.2	<i>Development</i>	92
4.5.3	<i>Testing</i>	101
4.5.4	<i>Review</i>	101
4.6	Iterasi 4.....	102
4.6.1	<i>Design</i>	102
4.6.2	<i>Development</i>	103
4.6.3	<i>Testing</i>	109
4.6.4	<i>Review</i>	110
4.7	Evaluasi Temuan Penelitian.....	110
4.8	Pengujian dan Kepuasan Pengguna	113
V.	PENUTUP	133
5.1	Kesimpulan	133
5.2	Saran.....	134
	DAFTAR PUSTAKA	135
	LAMPIRAN.....	138

DAFTAR TABEL

Tabel	Halaman
Tabel 1. Prinsip Pengembangan <i>Agile Development</i>	6
Tabel 2. Komponen dan Simbol <i>Use Case Diagram</i>	12
Tabel 3. Komponen dan Simbol <i>Activity Diagram</i>	13
Tabel 4. Tim Pengembang Platform Quision.....	20
Tabel 5. Waktu Pelaksanaan Penelitian	28
Tabel 6. Perangkat Keras	29
Tabel 7. Perangkat Lunak	29
Tabel 8. Data Hasil Kuesioner	35
Tabel 9. Evaluasi Pengguna	37
Tabel 10. Evaluasi Tim Internal.....	38
Tabel 11. Perbandingan Aspek Evaluasi.....	39
Tabel 12. Daftar Temuan Hasil Evaluasi	40
Tabel 13. Kebutuhan Fungsional	41
Tabel 14. Kebutuhan Non-Fungsional	42
Tabel 15. <i>User Stories</i>	42
Tabel 16. Aktor dan Deskripsi <i>Use Case</i>	45
Tabel 17. Rancangan Pelaksanaan Pengembangan.....	56
Tabel 18. <i>System Testing</i> Iterasi Ke-1.....	73
Tabel 19. <i>Review</i> Iterasi Ke-1	73
Tabel 20. <i>System Testing</i> Iterasi Ke-2.....	89
Tabel 21. <i>Review</i> Iterasi Ke-2	89
Tabel 22. <i>System Testing</i> Iterasi Ke-3.....	101
Tabel 23. <i>Review</i> Iterasi Ke-3	101
Tabel 24. <i>System Testing</i> Iterasi Ke-4.....	109
Tabel 25. <i>Review</i> Iterasi Ke-4	110
Tabel 26. Daftar Responden.....	113
Tabel 27. Variabel <i>Usability Testing</i>	114
Tabel 28. Daftar Tugas <i>Moderated Assessment Usability Testing</i>	115
Tabel 29. Daftar Hasil <i>Moderated Assessment Usability Testing</i>	115
Tabel 30. Frekuensi Umpan Balik <i>Moderated Assessment Usability Testing</i>	121
Tabel 31. Daftar Tugas <i>Moderated Comparative Usability Testing</i>	123
Tabel 32. Daftar Hasil <i>Moderated Comparative Usability Testing</i>	123

Tabel 33. Frekuensi Umpan Balik <i>Moderated Comparative Usability Testing</i> ...	128
Tabel 34. Daftar Pertanyaan Evaluasi Tingkat Kepuasan Pengguna.....	130
Tabel 35. Rekapitulasi Nilai Pertanyaan Evaluasi Tingkat Kepuasan Pengguna	131

DAFTAR GAMBAR

Gambar	Halaman
Gambar 1. Alur Kerja <i>Scrum</i> [5].....	8
Gambar 2. Tahapan <i>Scrum</i>	30
Gambar 3. <i>Use Case Diagram</i> Pengembangan Platform Quision	49
Gambar 4. <i>Activity Diagram</i> Melihat Informasi Promo.....	50
Gambar 5. <i>Activity Diagram</i> Memesan Responden.....	51
Gambar 6. <i>Activity Diagram</i> Membayar Pesanan.....	52
Gambar 7. <i>Activity Diagram</i> Mengakses Daftar Kuesioner Terbit.....	53
Gambar 8. <i>Activity Diagram</i> Mengakses Detail Kuesioner	54
Gambar 9. <i>Activity Diagram</i> Permintaan Pencairan Qu Coin.....	55
Gambar 10. Desain Halaman Detail Kuesioner (Peneliti)	57
Gambar 11. Desain Halaman Detail Kuesioner (Peneliti Detail)	58
Gambar 12. Desain Halaman Pilihan Produk	59
Gambar 13. Tampilan Halaman Detail Kuesioner (Peneliti)	59
Gambar 14. Tampilan Halaman Detail Kuesioner (Peneliti Detail)	60
Gambar 15. Tampilan Halaman Pilihan Produk	60
Gambar 16. Tampilan Halaman Produk Tambah Responden.....	61
Gambar 17. Tampilan Halaman Berhasil Tambah Responden.....	61
Gambar 18. Tampilan Halaman Detail Kuesioner (Responden)	62
Gambar 19. Tampilan Halaman Detail Kuesioner (Responden Konfirmasi)	63
Gambar 20. Kode Mengambil Data Chart Detail <i>Engagement</i> Kuesioner	64
Gambar 21. Kode Mengambil Data Detail <i>Engagement</i> Kuesioner	66
Gambar 22. Konfigurasi Produk Tambah Responden	67
Gambar 23. <i>Elementor</i> Halaman Detail Kuesioner.....	68
Gambar 24. Kode Pengecekan Peneliti Kuesioner	69
Gambar 25. Kode Visualisasi Data Chart Detail <i>Engagement</i> Kuesioner.....	70
Gambar 26. Kode <i>Autofill</i> Formulir Konfirmasi Pengisian Kuesioner.....	71
Gambar 27. Desain Halaman Kumpulan Kuesioner	74
Gambar 28. Desain Halaman Kumpulan Kuesioner (<i>Smart</i>).....	75
Gambar 29. Desain Halaman Kumpulan Kuesioner (Filter).....	75
Gambar 30. Tampilan Halaman Kumpulan Kuesioner.....	76
Gambar 31. Tampilan Halaman Kumpulan Kuesioner (<i>Smart</i>).....	76
Gambar 32. Tampilan Halaman Kumpulan Kuesioner (Filter)	77

Gambar 33. Kode Sisa Waktu Kuesioner	77
Gambar 34. Kode Pengecekan Kumpulan Kuesioner Bagian 1	78
Gambar 35. Kode Pengecekan Kumpulan Kuesioner Bagian 2	80
Gambar 36. Konfigurasi <i>Taxonomy</i>	81
Gambar 37. Konfigurasi Terbitkan Kuesioner Dengan <i>Taxonomy</i>	82
Gambar 38. Konfigurasi <i>Prefix</i> URL <i>Taxonomy</i>	83
Gambar 39. Kode Penerapan Filter	84
Gambar 40. Kode Penerapan Pencarian.....	86
Gambar 41. <i>Elementor</i> Halaman Kumpulan Kuesioner	87
Gambar 42. Kode Sidebar <i>Toggle</i>	88
Gambar 43. Desain Halaman Pesan Responden	90
Gambar 44. Desain Halaman Keranjang.....	91
Gambar 45. Desain Halaman <i>Checkout</i>	91
Gambar 46. Tampilan Halaman Pesan Responden	92
Gambar 47. Tampilan Halaman Detail Produk Responden	92
Gambar 48. Tampilan Halaman Keranjang	93
Gambar 49. Tampilan Halaman <i>Checkout</i>	93
Gambar 50. Kode Implementasi Fungsi Input Koin Pada Keranjang.....	94
Gambar 51. Kode Implementasi Fungsi Penerapan Koin Pada Keranjang	96
Gambar 52. Konfigurasi Produk Responden	97
Gambar 53. <i>Elementor</i> Halaman Keranjang	98
Gambar 54. <i>Elementor</i> Halaman <i>Checkout</i>	99
Gambar 55. Kode Implementasi Formulir Kupon Dan Penukaran Qu Coin	100
Gambar 56. Desain Halaman Promo.....	102
Gambar 57. Desain Halaman Pencairan Qu Coin.....	102
Gambar 58. Tampilan Halaman Promo	103
Gambar 59. Tampilan Halaman Pencairan Qu Coin.....	103
Gambar 60. <i>Elementor</i> Halaman Promo	104
Gambar 61. Kode Otomasi Penyalinan Kode Kupon	104
Gambar 62. Kode <i>Autofill</i> Formulir Pencairan	105
Gambar 63. Kode Formulir Pencairan	106
Gambar 64. <i>Elementor</i> Halaman Pencairan Qu Coin	108
Gambar 65. Konsep Halaman <i>Sandbox</i>	111
Gambar 66. Model <i>Version Control</i>	112

I. PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi serta ilmu pengetahuan tidak dapat dilepaskan dari kontribusi penelitian. Penelitian merupakan faktor penting dalam mengembangkan ilmu pengetahuan dan teknologi [1]. Data menjadi elemen krusial dalam setiap penelitian, dan instrumen penelitian seperti angket, tes, lembar observasi, serta wawancara merupakan alat bantu yang umum digunakan untuk mengumpulkan data kuantitatif [2][3]. Namun, salah satu tantangan utama dalam penelitian adalah sulitnya menemukan responden yang bersedia mengisi kuesioner, sehingga berpotensi memperpanjang durasi penelitian [4].

Di era digital saat ini, kuesioner dapat dengan mudah dibuat menggunakan berbagai platform digital seperti *Google Form*, *Typeform*, *Survey Monkey*, dan platform pembuat kuesioner lainnya. Metode penyebaran kuesioner yang umum dilakukan saat ini adalah dengan menyebarkan pada grup *WhatsApp*, media sosial, atau secara personal kepada calon responden. Namun, upaya penyebaran kuesioner ini belum sepenuhnya efektif karena banyak calon responden cenderung mengabaikan kuesioner. Peneliti yang mengalami masalah ini seringkali menggunakan jasa pengisian kuesioner yang tersedia di platform Instagram atau *marketplace*. Namun, solusi ini memiliki kelemahan seperti ketidakjelasan asal usul responden, ketidaksesuaian kriteria responden, dan potensi kecurangan seperti penggunaan *bot* atau *email* palsu. Akibatnya, keakuratan data yang dihasilkan patut dipertanyakan. Penting untuk mengetahui dengan jelas kondisi, situasi, dan jumlah responden yang akan menjadi sumber data untuk memastikan bahwa respon yang diperoleh adalah akurat dan representatif [2]. Oleh karena itu, dibutuhkan solusi yang lebih terpercaya untuk menghubungkan peneliti dengan responden.

Berdasarkan latar belakang permasalahan tersebut, muncul kebutuhan akan sebuah platform yang dapat menghubungkan peneliti dan responden. Analisis pasar menunjukkan bahwa solusi yang ada saat ini, seperti pada jasa isi kuesioner di media sosial dan *marketplace* masih memiliki keterbatasan pada akurasi data, asal usul responden, dan biaya yang cukup tinggi. Oleh karena itu, Quision hadir sebagai platform digital berbasis website yang bertujuan memudahkan peneliti dalam menemukan responden yang sesuai dengan kriteria penelitiannya. Melalui Quision, peneliti dapat mempublikasikan kuesionernya pada platform, sehingga responden dapat mengisi kuesioner tersebut dan mendapatkan imbalan yang dapat ditukar dengan uang. Quision tidak hanya mempercepat proses pengumpulan data, tetapi juga memastikan keakuratan dan relevansi data yang dikumpulkan. Pengembangan awal Quision telah berhasil membangun dasar platform yang solid, baik dari sisi *backend* maupun *frontend*. Namun, seiring dengan perkembangan kebutuhan pasar dan pengguna yang terus berubah, pengembangan lebih lanjut menjadi suatu urgensi. Meningkatnya permintaan mengharuskan adanya penyempurnaan dan optimalisasi fitur yang ada, serta penambahan fitur penunjang. Pengembangan ini mencakup berbagai aspek, mulai dari peningkatan sistem penyaringan responden, penguatan transparansi, promosi, hingga perluasan jenis layanan yang ditawarkan. Tanpa pengembangan berkelanjutan, Quision beresiko kehilangan daya saing dan tidak mampu menangani permintaan yang semakin besar dan beragam. Dengan menerapkan prinsip *agile*, Quision diharapkan dapat terus berevolusi dan beradaptasi dengan dinamika pasar serta kebutuhan pengguna. Evaluasi dan pengembangan berfokus pada penyempurnaan dan optimalisasi fitur yang ada serta penambahan fitur penunjang. Oleh karena itu, pengembangan lebih lanjut dari Quision menjadi langkah penting untuk memastikan keberlanjutan dan relevansi platform dalam mendukung penelitian di masa depan.

1.2 Rumusan Masalah

Adapun rumusan masalah pada penelitian ini adalah sebagai berikut:

1. Bagaimana mendapatkan kebutuhan pengguna terbaru berdasarkan pengembangan terdahulu?

2. Bagaimana menerapkan kebutuhan pengguna terbaru menggunakan metode *Agile*?

1.3 Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah sebagai berikut:

1. Mengevaluasi pengembangan terdahulu untuk mendapatkan kebutuhan pengguna terbaru.
2. Mengembangkan fitur pada platform Quision dengan menggunakan metode *Agile*.

1.4 Manfaat Penelitian

Adapun manfaat dari penelitian ini adalah sebagai berikut:

1. Penelitian ini berkontribusi pada pengembangan platform Quision yang dapat memfasilitasi peneliti memperoleh data dari responden secara cepat dan akurat.
2. Hasil penelitian ini dapat menjadi dasar bagi pengembangan lebih lanjut di masa depan.

1.5 Batasan Masalah

Adapun batasan masalah dalam penelitian ini adalah:

1. Pengembangan platform ini melanjutkan pengembangan terdahulu yang berbasis *open-source Content Management System (CMS)* yaitu *WordPress* dengan *plugins*, integrasi *payment gateway*, dan modifikasi fungsionalitas sistem tambahan untuk menunjang kebutuhan platform.
2. Penelitian ini dilakukan melalui kolaborasi antara tiga anggota tim, di mana anggota pertama mengembangkan aspek *backend* awal, anggota kedua menangani pengembangan aspek *frontend* awal, dan penulis berfokus pada pengembangan *frontend* dan *backend* fitur terbaru setelah pengembangan awal.
3. Penelitian ini hanya dilakukan sampai tahap pengujian fitur yang dikembangkan tanpa melakukan pengembangan lanjutan terkait hasil pengujian.

1.6 Sistematika Penulisan

Adapun struktur penulisan yang diterapkan pada penelitian ini sebagai berikut:

BAB I: PENDAHULUAN

Bab ini berisi latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, dan sistematika penulisan laporan.

BAB II: TINJAUAN PUSTAKA

Bab ini berisi tentang penelitian-penelitian sebelumnya dan teori-teori yang mendukung sebagai referensi dalam penelitian ini.

BAB III: METODOLOGI PENELITIAN

Bab ini berisi tentang waktu dan tempat penelitian, alat dan bahan yang digunakan dalam penelitian, dan prosedur serta analisis kebutuhan perangkat lunak.

BAB IV: HASIL DAN PEMBAHASAN

Bab ini berisi pembahasan mengenai penelitian yang dilakukan.

BAB V: KESIMPULAN DAN SARAN

Bab ini memuat kesimpulan berdasarkan hasil penelitian dan saran yang diharapkan untuk pengembangan penelitian selanjutnya.

DAFTAR PUSTAKA

LAMPIRAN

II. TINJAUAN PUSTAKA

2.1 Dasar Teori

2.1.1 *Agile Software Development*

Dalam beberapa tahun terakhir, metodologi pengembangan perangkat lunak *agile* telah mengalami peningkatan popularitas yang signifikan. Hal ini terjadi karena *agile* mampu memberikan solusi yang lebih adaptif dan responsif terhadap kebutuhan bisnis yang cepat berubah. Sebagai respons terhadap keterbatasan model tradisional seperti *waterfall*, *agile* menawarkan pendekatan yang lebih fleksibel dan iteratif dalam pengembangan perangkat lunak. Filosofi utama *agile* tercermin dalam *agile manifesto* yang menitikberatkan pada individu dan interaksi, perangkat lunak yang berfungsi, kolaborasi dengan pengguna, serta respons terhadap perubahan. Pendekatan ini memungkinkan tim pengembangan untuk terus menyempurnakan produk berdasarkan umpan balik pelanggan, sehingga meningkatkan kepuasan pengguna. Konsep dasar dari *agile* adalah untuk mendorong kolaborasi yang lebih baik antara tim pengembang dan pemangku kepentingan, sehingga memungkinkan umpan balik yang lebih cepat dan adaptasi terhadap perubahan kebutuhan selama proses pengembangan. Hal ini menjadikan *agile* pilihan yang populer dalam lingkungan yang serba cepat berubah, dinamis, dan tidak pasti. Berbagai metode *agile* meliputi *Extreme Programming (XP)*, *Scrum*, *Kanban*, *Lean Software Development*, *Feature-Driven Development (FDD)*, *Dynamic Systems Development Method (DSDM)*, dan *Crystal Methods*. Masing-masing metode ini memiliki ciri khas tertentu dalam penerapan prinsip *agile*. *Scrum* berfungsi sebagai kerangka manajemen proyek, *XP* fokus pada teknik rekayasa perangkat lunak, sementara *lean* mengedepankan efisiensi dan perbaikan

berkelanjutan. Setiap proyek dapat mengembangkan pendekatan *agile* sendiri, bergantung pada situasi dan tantangan yang dihadapi. Oleh karena itu, *agile development* mendorong tim untuk terus berpikir mandiri, fleksibel, dan inovatif dalam menerapkan prinsip-prinsipnya demi keberhasilan proyek. Prinsip-prinsip *agility* berjumlah dua belas dan menjadi landasan utama bagi siapa pun yang ingin menerapkan metode *agile* [5]. Berikut merupakan 12 prinsip pengembangan dengan metode *agile* (Tabel 1).

Tabel 1. Prinsip Pengembangan *Agile Development*

No	Prinsip
1	Prioritas tertinggi adalah kepuasan pelanggan melalui pengiriman perangkat lunak yang bernilai dan berkelanjutan.
2	Menyambut perubahan kebutuhan, bahkan di tahap akhir pengembangan.
3	Mengirimkan fungsional perangkat lunak secara berkala, dengan siklus singkat.
4	Kolaborasi harian antara pihak bisnis dan pengembang.
5	Membangun proyek di sekitar individu yang termotivasi, sehingga membuat lingkungan yang saling mendukung.
6	Komunikasi paling efektif adalah percakapan tatap muka.
7	Perangkat lunak yang berfungsi sebagai ukuran utama kemajuan.
8	Mendukung pengembangan berkelanjutan dengan kecepatan kerja stabil.
9	Perhatian terus-menerus terhadap keunggulan teknis dan desain yang baik.
10	Memaksimalkan pekerjaan, bukan jumlah tugas yang diselesaikan (kesederhanaan).
11	Arsitektur, kebutuhan, dan desain terbaik muncul dari tim yang mengorganisasi diri sendiri.
12	Tim secara berkala merefleksikan dan menyesuaikan diri untuk meningkatkan efektivitas.

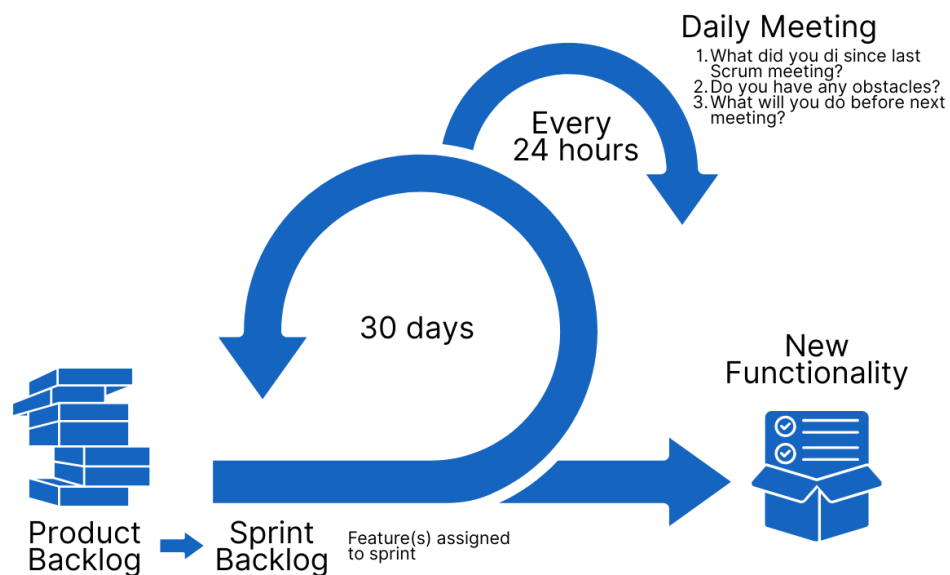
Salah satu elemen penting dalam *agile* adalah *master story list*, yaitu daftar seluruh *user stories* yang mencerminkan fitur-fitur yang diinginkan pengguna. Daftar ini diprioritaskan oleh pengguna, diestimasi oleh tim pengembang, dan menjadi landasan utama dalam perencanaan proyek. Proses pengembangan dilakukan dalam iterasi singkat berdurasi satu hingga dua minggu, di mana setiap iterasi bertujuan untuk mengubah *user story* prioritas tertinggi menjadi perangkat lunak yang berjalan dan telah diuji. Kapasitas kerja *agile* menggunakan konsep tim *velocity*, yaitu ukuran seberapa banyak pekerjaan yang dapat diselesaikan dalam satu iterasi. Dengan memantau *velocity* secara rutin, tim dapat membuat

perencanaan yang realistis dan menghindari komitmen yang berlebihan. Salah satu prinsip utama dalam *agile* adalah kemampuan beradaptasi terhadap perubahan. Ketika pekerjaan yang harus dilakukan melebihi kapasitas, tim diharapkan fleksibel dalam menyesuaikan ruang lingkup proyek agar tetap sesuai dengan kondisi nyata. *Agile* mengajarkan bahwa perubahan adalah sesuatu yang wajar dalam pengembangan perangkat lunak dan rencana harus selalu siap untuk diperbarui sesuai kebutuhan. Dalam prinsip *agile*, sebuah fitur baru dianggap "*done*" hanya jika sudah melewati seluruh tahapan pengembangan, mulai dari analisis, desain, *coding*, hingga pengujian, sehingga dapat dikatakan siap untuk dirilis. Hal ini menekankan pentingnya menghasilkan perangkat lunak yang benar-benar berfungsi sebagai ukuran utama keberhasilan. *Agile* juga mendasarkan pendekatannya pada tiga kebenaran sederhana dalam proyek pengembangan perangkat lunak. Pertama, tidak mungkin mengumpulkan seluruh kebutuhan di awal proyek. Kedua, kebutuhan yang telah dikumpulkan pasti akan berubah. Ketiga, selalu akan ada lebih banyak pekerjaan daripada waktu dan dana yang tersedia [6]. Dengan menerima ketiga kenyataan ini, tim dapat mengurangi stres dan meningkatkan fokus dalam menjalankan proyek.

2.1.2 Scrum

Scrum merupakan salah satu metodologi pengembangan perangkat lunak *Agile* yang paling populer dan efektif. Metode ini berfokus pada fleksibilitas, kolaborasi tim, dan *deployment* produk secara *incremental*. *Scrum* memungkinkan tim untuk beradaptasi dengan cepat terhadap perubahan kebutuhan dan prioritas proyek. Hal ini sangat penting dalam lingkungan pengembangan perangkat lunak yang dinamis. *Scrum* memiliki prinsip yang sejalan dengan *agile manifesto*, dengan kerangka kerja yang mencakup aktivitas *requirements*, *analysis*, *design*, *evolution*, dan *delivery*. Dalam setiap aktivitas ini, pekerjaan dilakukan dalam pola proses yang disebut *sprint*. Iterasi atau *sprint* merupakan inti dari proses kerja menggunakan dimana periodenya tetap (biasanya 1-4 minggu), di mana tim bekerja untuk menyelesaikan sekumpulan fitur atau kebutuhan yang sudah ditentukan dari *backlog*. *Backlog* sendiri merupakan daftar prioritas kebutuhan atau fitur produk.

Backlog bersifat dinamis dan dapat diperbarui kapan saja oleh *product owner* untuk menyesuaikan prioritas, namun perubahan tidak diperbolehkan saat iterasi berlangsung agar tim dapat bekerja dalam lingkungan yang stabil. *Scrum* menerapkan pertemuan harian singkat yang disebut *scrum meeting*. Dalam pertemuan yang biasanya berdurasi 15 menit ini, setiap anggota tim menjawab tiga pertanyaan utama yaitu apa yang telah dikerjakan sejak pertemuan terakhir, hambatan apa yang dihadapi, dan apa yang akan dikerjakan hingga pertemuan berikutnya. Seorang *scrum master* memimpin pertemuan ini untuk membantu mengidentifikasi masalah sejak dini dan mendorong terciptanya tim yang terorganisir. Setelah iterasi selesai, tim mengadakan demo untuk memperlihatkan fungsionalitas baru yang telah dikembangkan kepada tim dan pengguna. Meskipun tidak semua fitur produk ditampilkan dalam demo, fitur yang diselesaikan selama iterasi dapat langsung diuji dan dievaluasi. Dengan adanya demo ini, pengguna dapat memberikan umpan balik dan memungkinkan tim untuk menyesuaikan pengembangan berikutnya dengan kebutuhan yang lebih actual [5]. Melalui proses iteratif dan kolaboratif ini, *scrum* mendukung pengembangan perangkat lunak yang adaptif, efisien, dan berorientasi pada kebutuhan pengguna.



Gambar 1. Alur Kerja *Scrum* [5]

Dalam pengembangan perangkat lunak tradisional, salah satu metode yang sering digunakan adalah model *plan-driven*, seperti *waterfall*. Model ini mengedepankan perencanaan awal yang rinci, mulai dari analisis kebutuhan secara lengkap, desain, pembangunan, hingga pengujian yang dilakukan secara berurutan. Pendekatan ini sangat efektif untuk masalah yang terdefinisi dengan baik, bersifat prediktif, dan tidak mengalami banyak perubahan. Namun, dalam kenyataannya, sebagian besar pengembangan produk mengandung ketidakpastian, sehingga model *plan-driven* sering kali gagal memenuhi kebutuhan yang dinamis tersebut. Di sisi lain, *scrum* lebih sesuai untuk lingkungan yang penuh ketidakpastian. *Scrum* fokus pada adaptasi terhadap perubahan secara *real-time*, iterasi yang cepat, dan transparansi dalam proses kerja. *Scrum* mendorong penyelesaian produk secara bertahap berdasarkan fitur, bukan berdasarkan fase kerja seperti dalam model *waterfall*. Pendekatan ini memastikan bahwa di setiap akhir iterasi, produk memiliki sebuah *increment* yang benar-benar berfungsi dan terintegrasi dengan *increment* sebelumnya. Umpan balik diperoleh di setiap akhir iterasi, sehingga tim dapat melakukan adaptasi terhadap prioritas fitur berikutnya atau proses kerja yang digunakan. Jika ditemukan kekurangan, *scrum* memungkinkan perbaikan pada sprint-sprint berikutnya tanpa harus merencanakan ulang seluruh proses [7]. Oleh karena itu, *scrum* menjadi pendekatan yang sangat cocok untuk menghadapi kebutuhan pasar yang dinamis, karena memungkinkan bisnis beradaptasi secara cepat, memperbaiki kekurangan, dan terus mengembangkan produk yang relevan sesuai perubahan kebutuhan pengguna.

2.1.3 *User Stories*

User stories merupakan komponen kunci dalam metodologi pengembangan perangkat lunak *agile*, khususnya dalam kerangka kerja *Scrum*. *User stories* berfungsi sebagai deskripsi singkat dan sederhana tentang fitur atau fungsionalitas yang diinginkan dari perspektif pengguna akhir. *User stories* membantu tim pengembangan untuk memahami kebutuhan pengguna dengan lebih baik. Format umum *user story* adalah "Sebagai [peran pengguna], saya ingin [tujuan/keinginan] sehingga [manfaat]" [8]. Format ini memastikan bahwa setiap fitur yang

dikembangkan memiliki konteks yang jelas, tujuan yang spesifik, dan manfaat yang terukur. Dengan mengadopsi perspektif pengguna, tim pengembangan dapat lebih mudah mengidentifikasi dan memprioritaskan fitur-fitur yang penting bagi pengguna akhir.

User stories memainkan peran penting dalam proses *scrum*, terutama dalam tahap perencanaan dan eksekusi *sprint*. Selama *sprint planning*, tim memilih sejumlah *user stories* dari *product backlog* untuk dikerjakan dalam *sprint* yang akan datang. Pemilihan ini didasarkan pada prioritas yang ditetapkan oleh *product owner* dan kapasitas tim. *User stories* yang dipilih kemudian dipecah menjadi tugas-tugas yang lebih kecil dan dapat dikelola, sehingga memungkinkan tim untuk mengeksekusi kebutuhan dengan lebih efektif. Penggunaan *user stories* dalam *scrum* mendorong komunikasi yang lebih baik antara *stakeholder*, *product owner*, dan tim pengembang [8]. *User stories* menyediakan bahasa umum yang dapat dipahami oleh semua pihak, terlepas dari berbagai latar belakang teknis. Hal ini berdampak pada diskusi tentang kebutuhan proyek yang lebih aktif dan pengambilan keputusan yang lebih tepat.

2.1.4 *Front End*

Frontend development merupakan aspek penting dalam pengembangan aplikasi web modern yang berfokus pada pengalaman pengguna dan interaksi langsung dengan sistem. *Frontend developer* bertanggung jawab untuk mengimplementasikan desain visual dan memastikan fungsionalitas antarmuka pengguna. *Frontend developer* bekerja dengan teknologi seperti *HyperText Markup Language* (HTML), *Cascading Style Sheet* (CSS), dan *JavaScript* untuk menciptakan antarmuka yang responsif dan menarik. Responsivitas dan performa menjadi fokus utama dalam *frontend development*. *Frontend developer* harus memastikan bahwa aplikasi web dapat berjalan dengan baik di berbagai ukuran layar dan perangkat [9]. Optimisasi performa *frontend* menjadi fokus utama dalam pengembangan web modern. Teknik-teknik seperti *lazy loading* dan *code splitting* dapat diimplementasikan untuk meningkatkan kecepatan loading. Selain itu,

implementasi desain responsif dan adaptif menjadi standar untuk memastikan konsistensi tampilan dan fungsi pada berbagai perangkat dan ukuran layar.

2.1.5 *Back End*

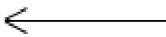
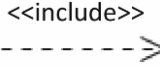
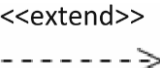
Backend development berfokus pada logika *server*, manajemen *database*, dan infrastruktur yang mendukung aplikasi web dan fungsionalitas *frontend*. *Backend developer* bertanggung jawab untuk memastikan keamanan, skalabilitas, dan kinerja aplikasi di sisi server. *Backend developer* bekerja dengan *server-side programming language*, sistem manajemen *database*, dan berbagai *tools server*. Keamanan merupakan prioritas utama dalam *backend development*. Pengembang harus menerapkan praktik-praktik keamanan terbaik, seperti enkripsi data, autentikasi yang kuat, dan perlindungan terhadap serangan umum seperti *SQL injection* dan *cross-site scripting* (XSS). Skalabilitas adalah aspek penting lainnya dalam *backend development*. Sistem *backend* harus dirancang supaya dapat menangani peningkatan beban dan pertumbuhan data [9]. Teknik-teknik seperti *caching*, *load balancing*, dan optimisasi *database* dapat diimplementasikan untuk meningkatkan kecepatan akses data dan mengurangi beban pada *server database*. Manajemen API (*Application Programming Interface*) juga menjadi komponen penting selanjutnya dalam pengembangan *backend* yang memungkinkan komunikasi efisien antara *frontend* dan *backend*, serta integrasi dengan layanan pihak ketiga. Pengembang backend harus merancang dan mengimplementasikan API yang aman, efisien, dan mudah digunakan.

2.1.6 *Use Case Diagram*

Use case diagram merupakan salah satu diagram dalam *Unified Modeling Language* (UML) yang digunakan untuk memodelkan interaksi antara sistem dan aktor-aktornya. Diagram ini sangat penting dalam tahap perancangan sistem informasi karena membantu menggambarkan fungsionalitas sistem dari perspektif pengguna. Hal yang ditekankan pada *use case diagram* terkait “apa” yang dapat dilakukan oleh sistem dan bukan “bagaimana” [10]. Penggunaan *use case diagram*

memungkinkan tim pengembang untuk memahami kebutuhan pengguna dengan lebih baik dan merancang sistem yang sesuai dengan ekspektasi mereka. *Use case diagram* terdiri dari beberapa komponen utama yang masing-masing memiliki simbol dan fungsi yang spesifik. Komponen-komponen pada *Use Case Diagram* akan dijelaskan pada tabel 2.

Tabel 2. Komponen dan Simbol *Use Case Diagram*

Simbol	Nama	Fungsi
	Aktor	Peran yang berinteraksi dengan sistem. Aktor dapat berupa manusia, organisasi, atau sistem eksternal.
	Use Case	Menggambarkan fungsi atau proses yang disediakan oleh sistem kepada aktor.
	System Boundary	Menandai batas antara sistem yang sedang dirancang dengan lingkungan eksternalnya.
	Asosiasi	Hubungan antara aktor dan use case karena terdapat interaksi antara keduanya.
	Generalisasi	Hubungan pewarisan yang melibatkan aktor atau use case satu dengan aktor atau use case lainnya.
	Include	Menandakan bahwa use case memasukkan fungsionalitas dari use case lain.
	Extend	Menunjukkan bahwa suatu use case dapat diperluas dengan fungsionalitas use case lain.

Penggunaan simbol-simbol dalam *use case diagram* membantu visualisasi interaksi antara pengguna dan sistem dengan cara yang jelas dan terstruktur [11]. Hal ini

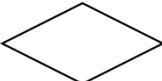
memudahkan berbagai pihak untuk memahami fungsionalitas sistem tanpa perlu mendalami detail teknis implementasinya.

2.1.7 Activity Diagram

Activity diagram merupakan suatu diagram UML yang menggambarkan konsep aliran data/kontrol, aksi terstruktur serta dirancang dengan baik dalam suatu sistem. *Activity diagram* sering digunakan untuk memvisualisasikan logika bisnis, alur kerja, dan aliran data dalam pengembangan perangkat lunak serta berbagai sistem bisnis lainnya. Diagram ini mengilustrasikan bagaimana sebuah proses bergerak dari satu aktivitas ke aktivitas lain, serta bagaimana keputusan dan percabangan diatur dalam alur kerja tersebut [12]. Selain itu, *activity diagram* juga dapat membantu mengidentifikasi ketidakefisienan, hambatan, atau potensi perbaikan dalam proses bisnis. Dengan representasi visual yang jelas, diagram ini memudahkan komunikasi antara pengembang, analis sistem, dan pemangku kepentingan lainnya.

Activity diagram mendukung visualisasi *workflow* yang mencakup tindakan-tindakan paralel dan berurutan, yang masing-masing dapat disesuaikan dengan berbagai kondisi atau keputusan dalam proses. Berikut ini adalah tabel simbol yang umum digunakan dalam *activity diagram* (Tabel 3).

Tabel 3. Komponen dan Simbol *Activity Diagram*

Simbol	Nama	Fungsi
	<i>Initial node</i> atau status awal	Menandai titik mulai suatu aktivitas, dilambangkan dengan lingkaran padat
	Aktivitas	Menggambarkan kumpulan aksi yang dilambangkan dengan persegi panjang dengan ujung tumpul.
	Percabangan	Memilah alur berdasarkan kondisi tertentu, dilambangkan dengan belah ketupat dan berisi kriteria keputusan.

	Alur kontrol	Menunjukkan urutan eksekusi.
	Penggabungan	Menyatukan alur paralel.
	Pemisahan	Memisahkan alur paralel.
	<i>Final node</i> atau status akhir	Menghentikan seluruh alur, dilambangkan dengan lingkaran berlapis.

Activity diagram efektif untuk memodelkan berbagai jenis proses. Dalam proses pembuatan *activity diagram*, langkah-langkah utamanya meliputi identifikasi aktivitas, aliran kontrol, aliran objek, serta keputusan-keputusan penting yang terjadi di dalam proses. Pembuatan diagram diawali dengan satu *initial node* yang diletakkan di bagian atas atau kiri atas, serta diakhiri dengan satu *final node* di bagian bawah atau kanan bawah diagram untuk menjaga kesederhanaan dan keterbacaan. Penyusunan diagram sebaiknya memperhatikan urutan logis eksekusi aktivitas dari kiri ke kanan atau atas ke bawah. Dalam praktiknya, terdapat lima langkah utama dalam membuat *activity diagram*, yaitu memilih proses bisnis yang akan dimodelkan berdasarkan definisi kebutuhan dan *use case*, mengidentifikasi aktivitas-aktivitas yang diperlukan, menentukan alur kontrol dan *node* keputusan, mengidentifikasi aliran objek yang diperlukan untuk mendukung logika proses, serta menggambar diagram dengan mempertimbangkan estetika dan kemudahan pemahaman [11]. Dengan mengikuti langkah-langkah ini, *activity diagram* dapat menjadi representasi visual yang efektif dari logika bisnis.

2.1.8 Hosting

Hosting adalah layanan yang memungkinkan situs web dan aplikasi online diakses secara publik dengan menyediakan *server* dan ruang penyimpanan di internet. Hosting memberikan pengguna kapasitas untuk menyimpan data dan mengelola konten digital di server yang terhubung ke internet, sehingga memungkinkan situs atau aplikasi tersebut dapat diakses oleh pengguna di seluruh dunia. Hosting dapat diibaratkan sebagai ruang sewa dalam pusat perbelanjaan, di mana penyedia hosting menyediakan infrastruktur seperti perangkat keras, jaringan, dan fasilitas lainnya untuk mendukung keberadaan situs web di internet. Secara teknis, hosting menyediakan aksesibilitas dan mendukung fungsi situs web dengan menawarkan infrastruktur *server* yang dilengkapi dengan berbagai fitur seperti manajemen domain, konfigurasi, dan pengelolaan file. Setiap penyedia hosting menetapkan kapasitas penyimpanan, kecepatan transfer data, serta fitur pengelolaan lainnya yang berbeda sesuai kebutuhan pengguna, yang dapat disesuaikan dengan tujuan dan jenis situs web atau aplikasi yang dikelola. Layanan hosting juga memungkinkan pengelolaan sumber daya secara terpusat dan efektif, memudahkan pengguna untuk menambahkan fitur tambahan seperti basis data, *email*, serta CMS seperti *WordPress*, yang umum digunakan dalam pengembangan situs web [13]. Dengan adanya layanan hosting, organisasi dan individu dapat menyimpan dan mengelola konten digital yang terhubung ke internet tanpa harus membangun infrastruktur server sendiri. Hosting juga memberikan kemudahan dalam melakukan backup data, perlindungan keamanan, dan peningkatan kapasitas penyimpanan secara fleksibel. Penggunaan hosting dalam pengembangan situs web sangat bermanfaat untuk memaksimalkan performa, keamanan, dan ketersediaan informasi yang disediakan kepada pengguna.

2.1.9 Black Box Testing

Black-box testing merupakan metode pengujian perangkat lunak yang berfokus pada evaluasi fungsionalitas sistem tanpa mempertimbangkan struktur internal atau kode program. Dalam pendekatan ini, penguji memeriksa bagaimana

sistem merespons berbagai masukan (*input*) dan menghasilkan keluaran (*output*) sesuai dengan spesifikasi yang telah ditentukan. Tabel pengujian menggunakan *black-box testing* meliputi skenario, hasil yang diharapkan, hasil pengujian, dan kesimpulan. Skenario pengujian dirancang berdasarkan fungsi-fungsi utama yang ada dalam perangkat lunak. Hasil yang diharapkan disusun sesuai dengan spesifikasi kebutuhan, sementara hasil pengujian mencatat bagaimana perangkat lunak merespon masukan yang diberikan. Kesimpulan digunakan untuk menentukan apakah perangkat lunak sukses memenuhi spesifikasi atau memerlukan perbaikan lebih lanjut. Pendekatan ini menempatkan penguji pada sudut pandang pengguna akhir, yang memfokuskan pada bagaimana sistem merespons input tertentu dan menghasilkan output yang diharapkan. Metode ini digunakan untuk mengidentifikasi kesalahan pada fitur utama sistem, seperti validasi input, output yang tidak sesuai, serta kegagalan dalam interaksi antar modul. Salah satu kelebihan dari *black-box testing* adalah kemampuannya untuk mengidentifikasi aspek-aspek yang tidak memenuhi spesifikasi kebutuhan dalam pengembangan perangkat lunak. Hal ini membuat *black-box testing* menjadi metode yang efektif dalam menemukan kesalahan atau ketidaksesuaian dalam fungsionalitas perangkat lunak. Namun, metode ini juga memiliki keterbatasan, yaitu pengujian tidak dapat dilakukan sepenuhnya karena penguji memiliki pengetahuan yang terbatas mengenai perangkat lunak yang diuji [14]. Kesimpulannya, metode *black-box testing* cocok untuk memastikan sistem bekerja sesuai dengan spesifikasi dan mengungkap kesalahan pada tingkat pengguna akhir.

2.1.10 Usability Testing

Usability testing merupakan metode evaluasi yang digunakan untuk mengukur seberapa mudah suatu aplikasi atau sistem dapat digunakan oleh pengguna. Metode ini bertujuan untuk mengidentifikasi masalah yang dihadapi pengguna, seperti kesulitan navigasi, kesalahan dalam penggunaan fitur, atau ketidakpuasan terhadap antarmuka [15]. *Usability testing* dapat dilakukan melalui berbagai pendekatan, termasuk *moderated* dan *unmoderated testing*. *Moderated testing* melibatkan kehadiran moderator yang memandu peserta selama sesi

pengujian, sementara *unmoderated testing* dilakukan tanpa moderator, dengan peserta mengikuti instruksi yang telah disiapkan sebelumnya. Selain itu, *usability testing* dapat dilaksanakan secara *remote* atau *in-person*. *Remote testing* memungkinkan pengujian dilakukan dari jarak jauh, seringkali menggunakan platform online, sedangkan *in-person testing* dilakukan secara tatap muka di lokasi tertentu. Tujuan *usability testing* dapat bervariasi, tergantung pada tahap pengembangan produk. *Explorative testing* digunakan pada tahap awal untuk mengeksplorasi kebutuhan pengguna dan ide desain baru, sementara *assessment testing* bertujuan untuk mengevaluasi kinerja produk yang sudah ada. *Comparative testing*, di sisi lain, digunakan untuk membandingkan dua atau lebih produk atau desain guna menentukan mana yang lebih mudah digunakan atau lebih disukai oleh pengguna. Dalam pelaksanaannya, *usability testing* melibatkan beberapa langkah kunci, termasuk perencanaan, pelaksanaan, analisis data, dan pelaporan. Pada tahap perencanaan, peneliti menentukan tujuan pengujian, memilih metode yang sesuai, dan merekrut peserta yang mewakili target pengguna. Selama pelaksanaan, peneliti mengamati perilaku peserta dan mencatat kesulitan atau masalah yang mereka hadapi. Jumlah pengguna yang diperlukan untuk pengujian sangat bergantung pada skala proyek. Untuk proyek yang tergolong kecil, minimal diperlukan 5 pengguna, sedangkan untuk proyek yang lebih besar dibutuhkan adalah 15 pengguna. Variabel yang diukur dalam *usability testing* mencakup *task success rate*, *task completion time*, *error rate*, dan *user satisfaction*. *Task success rate* mengukur persentase tugas yang berhasil diselesaikan oleh peserta, sementara *task completion time* mencatat waktu yang dibutuhkan untuk menyelesaikan tugas [15]. *Task success rate* dihitung menggunakan persamaan berikut:

$$\text{Success Rate} = \frac{(S + (PS \times 0,5))}{\text{Total Task}} \times 100\%$$

S merupakan jumlah keberhasilan penuh dan PS merupakan keberhasilan sebagian. *Time Based Efficiency (TBE)* mengukur kecepatan pengguna dalam menyelesaikan tugas dan dihitung dengan persamaan:

$$TBE = \frac{\sum_{j=1}^R \sum_{i=1}^N \frac{n_{ij}}{t_{ij}}}{N \times R}$$

N = Jumlah total tugas

R = Jumlah total pengguna

n_{ij} = Hasil tugas 'i' oleh pengguna 'j'. Jika pengguna telah menyelesaikan tugas, maka $n_{ij} = 1$, jika tidak, maka $n_{ij} = 0$

t_{ij} = Waktu pengguna 'j' menyelesaikan tugas 'i'. Jika tugas tidak selesai, waktu diukur hingga pengguna berhenti mengerjakan tugas tersebut. *Error rate* mengukur tingkat kesalahan pengguna, dihitung dengan persamaan:

$$Error Rate = \frac{Total Defects}{Total Opportunities}$$

Total Defects merupakan total kesalahan yang dilakukan pengguna, dan *Total Opportunities* adalah total peluang melakukan kesalahan. *Error rate* dapat dikategorikan kecil atau wajar jika nilai rata-ratanya adalah sebesar 0,70. Selain itu, *usability testing* juga dapat menggunakan metrik *Single Ease Question* (SEQ). SEQ merupakan metrik yang membantu menilai seberapa mudah setiap tugas diselesaikan oleh pengguna. Metrik ini diberikan segera setelah tugas selesai dan hanya terdiri dari satu pertanyaan dengan skala penilaian 7 poin:

“Secara keseluruhan, seberapa mudahkah tugas tersebut diselesaikan?”

(1 = sangat sulit, 7 = sangat mudah)

Adapun perhitungan nilai evaluasi tingkat kepuasan pengguna berdasarkan metode SEQ dapat dihitung dengan persamaan berikut.

$$Nilai Akhir = \frac{\sum_{i=1}^{10} \left(\frac{Q1i+Q2i+Q3i+Q4i+Q5i+Q6i}{6} \right)}{10}$$

Metrik ini membantu mengevaluasi seberapa mudah atau sulit bagi pengguna untuk melakukan tindakan tertentu pada website. Nilai rata-rata SEQ dengan Tingkat kemudahan rendah adalah ≤ 5 [16]. Dengan demikian, usability testing tidak hanya membantu mengidentifikasi masalah, tetapi juga memberikan rekomendasi untuk meningkatkan desain agar lebih mudah digunakan dan memuaskan pengguna.

2.2 Pengembangan Terdahulu

Pengembangan terdahulu mengenai pengembangan platform Quision telah membahas aspek-aspek penting pembangunan awal *frontend* dan *backend*. Dalam penelitian tersebut, pendekatan *Agile Development* digunakan untuk memastikan pengembangan berjalan secara iteratif dan sesuai dengan kebutuhan pengguna. Pada tahap awal pengembangan platform Quision, penelitian sebelumnya telah melakukan analisis mendalam terhadap berbagai *tools* dan ekstensi yang dapat mendukung pengembangan. *WordPress* dipilih sebagai *framework* utama untuk pengembangan platform ini karena memiliki keunggulan dalam hal popularitas, kemudahan penggunaan, serta ketersediaan ekstensi yang luas. *WordPress* dilengkapi dengan *plugin-plugin* yang relevan, seperti *Ultimate Member* untuk manajemen anggota dan akses pengguna, *WooCommerce* untuk mendukung transaksi dan pemesanan jasa, dan *Contact Form* untuk pembuatan formulir serta interaksi pengguna. Analisis terhadap performa dan fungsionalitas plugin ini memastikan bahwa platform Quision dapat beroperasi dengan baik dan memenuhi kebutuhan pengguna dari berbagai segmen, baik peneliti maupun responden. *Elementor* dipilih karena menyediakan editor visual *drag-and-drop* yang memungkinkan pengguna membuat dan menyesuaikan halaman web dengan cepat serta memiliki ribuan *template* yang tersedia dan integrasi plugin yang luas, meskipun dapat mempengaruhi kinerja halaman dengan banyaknya fitur. Selain itu, Brevo dipilih sebagai alat untuk mengelola email karena fitur-fiturnya yang mendukung pengiriman dan pelacakan email, serta kemampuannya dalam pemasaran email dan transaksi. Pengembangan platform Quision melibatkan tiga anggota tim pengembang. Berikut tim pengembang platform Quision (Tabel 4).

Tabel 4. Tim Pengembang Platform Quision

No	Nama	Peran
1	Bella Amanda Iswahyudi	<i>Product Owner</i>
2	Raihan Ferdyanza	<i>Scrum Master, Back-end Developer</i>
3	Hafizh Zaghlul Rizal	<i>Front-end Developer</i>
4	Afrizal Yogi Pratama	<i>Full-stack Developer</i>

Pada sisi *frontend*, pengembangan sebelumnya yang dilakukan oleh anggota dua menekankan pada pengembangan antarmuka yang responsif dan mudah digunakan. Antarmuka pengguna yang dikembangkan difokuskan untuk mendukung interaksi yang intuitif dan efisien, sehingga pengguna dapat dengan mudah menavigasi platform. Pengembangan *frontend* juga mencakup pembuatan halaman-halaman penting lainnya, seperti halaman pemesanan jasa, halaman bantuan, dan panduan alur sistem. Fitur utama seperti halaman pendaftaran, login, dan dashboard pengguna menjadi fokus utama pada tahap ini. Halaman pendaftaran dan login memungkinkan pengguna baru untuk membuat akun dan pengguna terdaftar untuk mengakses platform, sedangkan dashboard menyediakan akses ke berbagai fitur seperti pengelolaan kuesioner dan riwayat aktivitas.

Pada sisi *backend*, pengembangan sebelumnya yang dilakukan oleh anggota satu fokus pada pengembangan fitur-fitur inti yang mendukung operasional platform. Pengelolaan data pengguna, pemrosesan pemesanan jasa, serta integrasi dengan berbagai API eksternal merupakan beberapa aspek penting yang dikembangkan. Pengembangan *backend* juga mencakup implementasi mekanisme pembayaran yang aman dan transparan, mendukung transaksi baik peneliti maupun responden, serta memastikan bahwa semua proses berjalan sesuai dengan standar fungsionalitas dan keamanan yang telah ditetapkan. Penelitian ini juga memperhatikan aspek pengelolaan kuesioner, di mana *backend* mendukung fitur-fitur seperti penerbitan kuesioner, pengelolaan data responden, dan pelaporan hasil survei. *Backend* yang dirancang mampu menangani permintaan responden untuk peneliti dengan efisien dan memiliki nilai ekonomi. Dengan demikian, penelitian sebelumnya telah menetapkan fondasi yang kuat bagi pengembangan platform Quision dengan 6 iterasi, baik dari sisi *frontend* maupun *backend*, dengan

memanfaatkan *tools* dan ekstensi yang telah dianalisis dengan cermat untuk mendukung kebutuhan pengguna. Penelitian yang akan peneliti bahas akan lebih difokuskan pada penambahan fitur-fitur baru dan peningkatan yang belum dibahas dalam pengembangan terdahulu.

2.3 Tools yang Digunakan

2.3.1 Content Management System (CMS)

Content Management System (CMS) merupakan platform yang dirancang untuk memudahkan pengguna dalam mengelola dan memodifikasi konten pada situs website tanpa memerlukan pemahaman mendalam pada teknis pemrograman. Hal ini membuat CMS menjadi solusi yang efisien bagi individu atau organisasi yang ingin mengelola konten web secara mandiri tanpa harus memahami kode pemrograman atau untuk mempersingkat proses pengembangan suatu website. CMS telah menjadi alat yang sangat populer dalam pengembangan web karena kemampuannya untuk menyederhanakan proses manajemen konten. Beberapa contoh CMS yang umum digunakan antara lain *WordPress*, *Joomla*, dan *Drupal* untuk portal web, serta *Prestashop*, *Opencart*, dan *Magento* untuk situs *e-commerce*. Dengan berbagai pilihan CMS yang tersedia, pengguna dapat memilih platform yang paling sesuai dengan kebutuhan mereka, baik untuk situs web sederhana maupun kompleks. *WordPress* sangat dikenal karena kemudahan penggunaannya dan fleksibilitasnya dalam menangani berbagai jenis situs web, mulai dari blog pribadi hingga situs perusahaan besar. Sementara itu, *Joomla* dan *Drupal* menawarkan fitur yang lebih kompleks dan dapat disesuaikan, menjadikannya pilihan yang baik untuk situs web dengan kebutuhan yang lebih spesifik [17]. Keunggulan utama CMS adalah kemampuannya untuk memisahkan konten dari desain dan fungsionalitas situs web. Hal ini memungkinkan pengembang untuk fokus pada pembuatan dan pengelolaan konten tanpa harus khawatir tentang aspek teknis seperti pengkodean atau desain tampilan. Selain itu, CMS juga menyediakan berbagai plugin dan modul yang dapat ditambahkan untuk meningkatkan fungsionalitas situs web, keamanan, dan integrasi dengan platform

pihak ketiga. Dengan demikian, CMS tidak hanya memudahkan pengelolaan konten tetapi juga memungkinkan pengguna untuk mengembangkan situs web yang lebih dinamis dan interaktif.

2.3.2 *WordPress*

WordPress merupakan salah satu *Content Management System* (CMS) terpopuler bersifat *open source* yang digunakan untuk membuat dan mengelola konten di internet. Berbasis PHP dan *MySQL*, platform ini awalnya dikembangkan pada tahun 2003 oleh Matt Mullenweg dan Mike Little sebagai media blog. Seiring perkembangan kebutuhan pengguna, *WordPress* bertransformasi menjadi sistem manajemen konten yang komprehensif dan kini menggerakkan lebih dari 40% situs web di internet [18]. *WordPress* terus berkembang menjadi sistem manajemen konten yang lebih fleksibel dan dapat digunakan untuk berbagai jenis website, mulai dari situs pribadi, portofolio, toko online, hingga website Perusahaan.

WordPress memiliki beragam fitur yang mendukung pengelolaan konten digital secara efisien. Beberapa fitur utama *WordPress* meliputi sistem *template* dan tema, plugin, sistem manajemen pengguna, artikel, halaman, serta desain responsif untuk perangkat *mobile*. Kemudahan penggunaan dan fleksibilitas menjadi faktor utama yang membuat *Wordpress* menjadi populer. Plugin memungkinkan integrasi fitur tambahan sesuai kebutuhan pengguna. Plugin yang dapat diintegrasikan seperti kode snippet, *payment gateway*, pengatur elemen, dan masih banyak lagi. Salah satunya, kode snippet merupakan blok kode yang dapat digunakan kembali (*reusable code*) dan digunakan untuk menyisipkan bagian kode kecil pada kode utama, sehingga dapat menambah fungsionalitas aplikasi tanpa penulisan ulang kode dari awal. *WordPress* tersedia dalam dua versi, yaitu *WordPress.org* (*self-hosted*) yang dapat diunduh dan diinstal pada hosting web pribadi dengan kontrol penuh, serta *WordPress.com* (*hosted*) yang merupakan layanan yang mendukung plugin dan tema berbayar maupun gratis dengan beberapa batasan dalam kustomisasi [17]. Dengan *WordPress.org*, pengembang memiliki kontrol penuh terhadap setiap aspek pada website dan dapat dengan fleksibel diterbitkan pada provider hosting atau server manapun. Selain itu, *WordPress* juga memiliki

dokumentasi dan komunitas online yang besar, sehingga mempermudah ketika mengalami masalah.

2.3.3 *Application Programming Interface (API)*

Application Programming Interface (API) merupakan komponen dalam pengembangan web modern yang memungkinkan aplikasi dan sistem berkomunikasi dan bertukar data. API dapat dilihat sebagai layanan perantara antara dua program yang menentukan bagaimana kedua program berinteraksi satu sama lain menggunakan permintaan (*request*) dan tanggapan (*response*). API berfungsi sebagai jembatan antara *front-end* dan *back-end*. API sering digunakan dalam pengembangan web untuk memasukan fitur pihak ketiga seperti seperti autentikasi pengguna, pembayaran *online*, atau layanan peta ke dalam situs web tanpa perlu membangun sistem dari awal. Cara kerja API yaitu program atau aplikasi web (klien) mengirim permintaan ke server dan server akan mengirimkan tanggapan. Dalam penggunaannya, API melibatkan berbagai metode komunikasi seperti protokol HTTP (*Hypertext Transfer Protocol*) dan HTTPS (*Hypertext Transfer Protocol Secure*), serta format data umum seperti JSON (*JavaScript Object Notation*) atau XML (*eXtensible Markup Language*) untuk melakukan pertukaran informasi. Dokumentasi API membantu pengembang memahami cara API digunakan yang mencakup parameter yang diperlukan, format permintaan, dan contoh respon yang diterima [19]. API memiliki keunggulan dalam pengembangan web yang memungkinkan tim pengembang saling bekerja sama dan memungkinkan pengembang menggunakan fitur yang sudah ada tanpa perlu membangunnya dari awal. Selain itu, API memungkinkan pengembangan aplikasi yang lebih modular dan terstruktur. Namun, API juga memiliki tantangan seperti keamanan, kompatibilitas versi, dan ketergantungan pada layanan pihak ketiga.

2.3.4 *MySQL*

My Structured Query Language (MySQL) merupakan salah satu manajemen basis data relasional (RDBMS) yang paling terkenal dan banyak

digunakan oleh pengembang dan administrator database karena kemampuannya untuk mengelola data dengan mudah, efisien, skalabel, dan aman. *Structured Query Language* untuk mengelola dan memanipulasi data membuat MySQL mudah dipelajari dan digunakan oleh pemula maupun profesional. MySQL menggunakan model *client-server*, dimana klien dapat terhubung dengan *server* MySQL melalui jaringan untuk mengakses dan mengubah data dalam database. Server MySQL berfungsi sebagai pengelola basis data yang menerima dan menjalankan perintah SQL dan kemudian mengirimkan hasilnya kembali pada klien. Salah satu keunggulan utama MySQL adalah kemampuannya dalam menangani volume data yang besar dengan kecepatan yang tinggi. MySQL juga dikenal dapat digunakan pada berbagai bahasa pemrograman, seperti PHP, *Python*, dan *Java* [20]. Hal ini membuat MySQL menjadi pilihan utama untuk pengembangan aplikasi web dan *mobile*. MySQL dapat dipasang dan dijalankan pada banyak platform seperti Windows, Linux, MacOS, platform cloud, framework seperti *Ruby*, *Laravel*, *CodeIgniter*, serta CMS seperti *Drupal*, *Joomla*, dan *WordPress*. Selain itu, MySQL memiliki komunitas pengembang yang besar dan aktif, yang menyediakan banyak sumber daya, tutorial, dan dukungan untuk pengguna baru.

2.3.5 Draw.io

Draw.io merupakan alat pembuatan diagram dan model gratis yang memiliki antarmuka sederhana dan intuitif. Draw.io menyediakan banyak bentuk dan elemen yang telah ditentukan sebelumnya, yang memudahkan pengguna dalam membuat diagram seperti *Use Case Diagram*, *Activity Diagram*, *Sequence Diagram*, dan diagram lainnya tanpa perlu memulai dari awal. Selain itu, draw.io terintegrasi dengan platform *cloud* populer seperti *Google Drive* dan *OneDrive*, sehingga memungkinkan pengguna untuk menyimpan dan berbagi diagram dengan mudah [21]. Keunggulan utama draw.io adalah kemampuannya untuk digunakan secara online tanpa perlu menginstal perangkat lunak tambahan, serta dukungannya untuk berbagai format file ekspor, seperti PNG, JPEG, dan PDF.

2.3.6 Figma

Figma merupakan alat desain berbasis website yang banyak digunakan untuk membuat, berbagi, dan menguji desain situs web, aplikasi, dan produk digital lainnya. Figma telah menjadi pilihan utama bagi banyak desainer antarmuka karena kemampuannya untuk menyederhanakan proses desain dan memfasilitasi kolaborasi antar tim. Keunggulan utama Figma adalah kemampuannya untuk memfasilitasi kolaborasi tim secara *real-time*, sehingga memungkinkan beberapa pengguna bekerja pada proyek yang sama secara bersamaan. Figma biasa digunakan untuk membuat desain awal seperti *wireframes* yang juga dikenal dengan *low-fidelity*, sampai pada tahap *prototyping* atau *high-fidelity*. Figma menawarkan berbagai fitur yang sangat berguna untuk desain antarmuka, salah satunya adalah komponen yang dapat digunakan kembali (*reusable components*). Hal ini tidak hanya memastikan konsistensi desain tetapi juga menghemat waktu dalam proses desain, serta mendukung pembuatan sistem desain (*design systems*) yang memungkinkan tim untuk mendefinisikan gaya, warna, tipografi, dan komponen antarmuka. Figma juga menyediakan berbagai plugin yang dapat membantu dalam proses desain [22]. Figma memungkinkan pembuatan prototipe yang interaktif dan mudah diuji. Hal ini sangat penting untuk memastikan bahwa desain telah memenuhi kebutuhan pengguna. Figma tidak hanya memudahkan proses desain tetapi juga memastikan bahwa hasil akhirnya sesuai dengan prinsip-prinsip *User Interface (UI)/User Experience (UX)* yang baik.

2.3.7 Hypertext Preprocessor (PHP)

Hypertext Preprocessor (PHP) merupakan bahasa skrip *server-side* yang banyak digunakan untuk membuat website dinamis. PHP dibuat pertama kali oleh Rasmus Lerdorf tahun 1994 yang sampai saat ini telah menjadi salah satu Bahasa pemrograman populer untuk pengembangan website karena kemudahan, fleksibilitas, dan kemampuannya berintegrasi dengan berbagai sistem manajemen basis data. PHP dirancang untuk menjadi bagian dalam kode HTML. PHP mendukung berbagai standar dan protokol website seperti HTTP, FTP, dan XML. Selain itu, PHP memiliki komunitas pengembang yang besar dan aktif yang

membantu mengembangkan *framework* dan *library* seperti *Laravel*, *Symfony*, dan *CodeIgniter*. PHP biasanya digunakan dalam membuat sistem manajemen konten (CMS) seperti *WordPress*, *Joomla*, dan *Drupal*, yang digunakan untuk membuat situs web dengan mudah tanpa pengetahuan pemrograman yang mendalam. Meskipun PHP memiliki banyak hak positif, ada beberapa masalah yang harus diperhatikan. Salah satunya adalah masalah keamanan karena PHP dapat terkena serangan seperti *SQL injection*, *cross-site scripting (XSS)*, dan *remote code execution* jika tidak dikonfigurasi dengan benar. Oleh karena itu, penerapan metode pengembangan yang aman, seperti penggunaan autentikasi, validasi input, dan sanitasi data sangatlah penting [23]. Secara keseluruhan, PHP merupakan bahasa pemrograman yang kuat dan fleksibel yang terus berkembang untuk memenuhi kebutuhan pengembangan web, sehingga menjadi salah satu pilihan bagi pengembang web di seluruh dunia karena dukungan komunitas yang besar, integrasi yang luas dengan teknologi lain, dan peningkatan yang dilakukan.

2.3.8 Chart.js

Chart.js merupakan *library JavaScript open-source* yang digunakan untuk membuat visualisasi data interaktif pada website. Chart.js dirancang untuk memudahkan pengembang dalam membuat grafik dan diagram yang menarik dengan cara yang sederhana dan efisien. Chart.js mendukung berbagai jenis grafik, termasuk grafik batang (*bar chart*), grafik garis (*line chart*), grafik lingkaran (*pie chart*), dan grafik radar (*radar chart*), yang dapat disesuaikan sesuai kebutuhan pengguna. Salah satu keunggulan utama Chart.js adalah kemudahan penggunaannya dengan hanya menuliskan beberapa baris kode untuk membuat grafik yang responsif dan interaktif. Chart.js juga mendukung animasi, yang membuat visualisasi data lebih menarik dan dinamis. Selain itu, *library* ini kompatibel dengan berbagai browser modern, sehingga memastikan bahwa grafik yang dibuat dapat diakses oleh pengguna di berbagai platform. Chart.js juga menyediakan dokumentasi yang lengkap dan komunitas yang aktif, yang memudahkan pengembang untuk mempelajari dan memecahkan masalah yang mungkin timbul selama penggunaan. Chart.js sangat cocok untuk proyek-proyek yang memerlukan visualisasi data yang cepat dan efektif, baik untuk aplikasi web

sederhana maupun kompleks [24]. Secara keseluruhan, Chart.js adalah alat yang sangat berguna untuk membuat visualisasi data interaktif di web. Dengan kemudahan penggunaan, dukungan untuk berbagai jenis grafik, dan kompatibilitas yang luas, Chart.js menjadi pilihan yang cocok bagi banyak pengembang dalam menampilkan data secara visual.

III. METODOLOGI PENELITIAN

3.1 Waktu dan Tempat Penelitian

3.1.1 Waktu Penelitian

Penelitian ini membahas pengembangan platform setelah platform beroperasi dan dapat dipasarkan. Penelitian terdahulu telah dilaksanakan selama kurang lebih tiga bulan atau 6 iterasi dengan total 30 *user stories*, sehingga iterasi 1 pada yang akan dilakukan penulis merupakan iterasi 7 pada keseluruhan pengembangan. Pembagian waktu penelitian dapat dilihat pada tabel 5.

Tabel 5. Waktu Pelaksanaan Penelitian

No	Kegiatan Penelitian	2024			2025		
		Okt	Nov	Des	Jan	Feb	Mar
1	Requirements						
2	Sprint Planning						
3	System Development						
4	System Testing						
5	Sprint Review						
6	Deployment						
7	Maintenance						
8	Penulisan Karya Ilmiah						

3.1.2 Tempat Penelitian

Penelitian ini dilakukan pada Laboratorium Teknik Digital, yang berada di bawah Program Studi Teknik Informatika, Jurusan Teknik Elektro, Fakultas Teknik, Universitas Lampung, dengan lokasi di Gedung Meneng, Kecamatan Rajabasa, Kota Bandar Lampung, Lampung 35141.

3.2 Alat dan Bahan Penelitian

3.2.1 Alat Penelitian

Adapun alat yang digunakan dalam penelitian dibagi menjadi dua jenis, yaitu perangkat keras (Tabel 6) dan perangkat lunak (Tabel 7).

Tabel 6. Perangkat Keras

No	Perangkat Keras	Spesifikasi	Kegunaan
1	Laptop	Intel i5-1135G7, RAM 16GB, Windows 11	Pengembangan dan pengujian website
2	<i>Smartphone</i>	Android 12	Pengujian website pada perangkat <i>mobile</i>

Tabel 7. Perangkat Lunak

No	Perangkat Lunak	Versi	Keterangan
1	Google Chrome (<i>Browser</i>)	Versi 128	<i>Tools</i> implementasi dan testing website
2	<i>Visual Studio Code</i>	Versi 1.82	<i>Text editor</i> untuk pengembangan kode
3	cPanel	Versi 120	Pengelolaan hosting
4	WordPress	Versi 6.4.5	<i>Tools</i> pengembangan website

5	PHP	Versi 8	Bahasa pemrograman pengembangan web
6	MySQL	Versi 8	Database untuk penyimpanan data

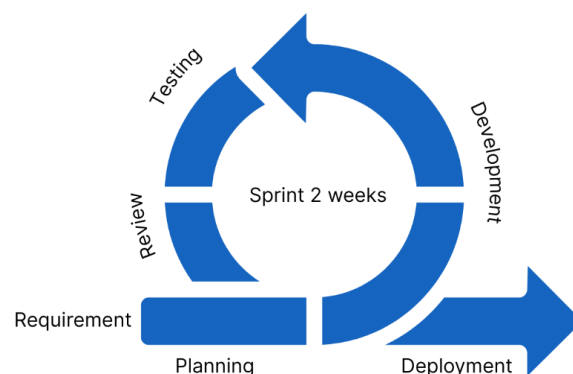
3.2.2 Bahan Penelitian

Penelitian ini memanfaatkan beberapa sumber sebagai bahan referensi untuk memperkuat pemahaman, teknik, dan langkah-langkah dalam pengembangan platform Quision, diantaranya adalah sebagai berikut:

1. Studi mengenai pengembangan fitur menggunakan CMS *WordPress*.
2. Analisis terhadap pengembangan terdahulu sebagai acuan dalam merancang pengembangan lanjutan.
3. Jurnal penelitian yang dijadikan referensi untuk metode dalam pengembangan platform.

3.3 Metode Penelitian

Penelitian ini bertujuan untuk mengembangkan platform Quision berbasis website untuk memudahkan proses penelitian dan pengumpulan data. Metode penelitian yang digunakan adalah *Agile Methodology* dengan kerangka kerja *Scrum*. Tahapan *Scrum* meliputi *Requirements*, *Planning*, *Development*, *Testing*, *Review*, *Deployment*, dan *Maintenance* (Gambar 2).



Gambar 2. Tahapan *Scrum*

Dalam setiap tahapan, tugas dipecah menjadi iterasi-iterasi pendek yang disebut *sprint*. Setiap *sprint* berlangsung antara satu hingga dua minggu dan bertujuan untuk menghasilkan peningkatan fungsionalitas yang dapat diuji dan dievaluasi. Pendekatan ini memungkinkan tim untuk mendapatkan umpan balik secara cepat dan melakukan penyesuaian yang diperlukan sebelum melanjutkan ke *sprint* berikutnya.

3.3.1 *Requirements*

Tahap *requirements* merupakan tahap untuk membuat daftar kebutuhan yang akan dikembangkan. Penulis akan melakukan observasi terhadap pengguna melalui umpan balik, baik peneliti maupun responden untuk memahami kebutuhan mereka. Selain itu, dilakukan evaluasi internal tim untuk meningkatkan kualitas dan pengalaman pengguna pada platform. Demi memastikan bahwa platform yang dikembangkan memenuhi ekspektasi, penulis akan memulai dengan mengidentifikasi kebutuhan fungsional dan non-fungsional. Kebutuhan fungsional adalah fitur-fitur inti yang harus ada pada platform agar dapat memenuhi tujuan yang telah ditetapkan, sedangkan kebutuhan non-fungsional mencakup aspek-aspek yang berkaitan dengan kualitas layanan platform, seperti kecepatan, skalabilitas, dan *user experience*. Dengan menggabungkan hasil observasi, evaluasi internal, dan analisis kebutuhan yang telah diidentifikasi, penulis akan menyusun dokumentasi menggunakan *Unified Modeling Language* (UML). *Use case* diagram digunakan untuk mengidentifikasi interaksi antara pengguna (aktor) dengan sistem dan *activity diagram* digunakan untuk memodelkan alur kerja atau proses yang terlibat dalam penggunaan suatu *use case*. Selanjutnya, *user stories* akan ditambahkan untuk memberikan konteks lebih rinci mengenai kebutuhan pengguna.

3.3.2 *Planning*

Tahap *planning* merupakan tahapan pengembangan platform untuk merencanakan *backlog* (daftar pekerjaan) untuk dikerjakan pada *sprint-sprint* mendatang. Fitur-fitur ini ditentukan dengan membagi *user stories* yang telah

dibuat sebelumnya menjadi beberapa kelompok. Penulis menggunakan metode *story point* untuk memperkirakan waktu yang dibutuhkan dalam mengembangkan setiap fitur dalam iterasi. Setiap 1 *story point* diartikan sebagai 1 hari pengerjaan. Pemilihan fitur dilakukan dengan mempertimbangkan skala prioritas, keterkaitan fungsional, dan bobot pekerjaan yang sudah ditetapkan. Pengelompokan fitur-fitur tersebut akan disesuaikan dengan jumlah iterasi, di mana setiap iterasi berlangsung antara satu hingga dua minggu.

3.3.3 *Development*

Tahap *development* adalah proses implementasi daftar tugas yang telah dirancang pada tahap sebelumnya. Penulis akan memulai dengan merancang algoritma fitur lalu memilih *tools* dan ekstensi yang tepat untuk menerapkan logika tersebut ke dalam kode. Tampilan antarmuka akan dikembangkan menggunakan plugin Elementor dan fungsionalitas fitur akan dikembangkan menggunakan kode *snippet*. Selanjutnya, penulis akan memastikan kode yang telah dibuat menerapkan *clean code*, mudah dipelihara, dan efektif. Antarmuka *frontend* dikembangkan menggunakan Elementor untuk memastikan responsivitas dan kemudahan penggunaan. Pada *backend*, dasbor admin *WordPress* disesuaikan untuk mengelola data dan berbagai fungsi lainnya. Fitur lain yang tidak tersedia secara *default*, pengembangan dilakukan menggunakan kode *snippet* PHP dan *JavaScript* untuk membuat fungsi khusus. Proses pengerjaan akan terus mengacu pada waktu yang telah ditetapkan untuk memastikan bahwa pengembangan tidak melebihi batas waktu yang telah ditetapkan.

3.3.4 *Testing*

Tahap *testing* merupakan bagian penting dari setiap *sprint* dalam *agile methodology*. Di sini, penulis akan melakukan pengujian fungsionalitas fitur yang telah dikembangkan. Pengujian diawali dengan *black box testing*, dimana tim pengembang menguji fungsionalitas fitur untuk memastikan bahwa fitur sudah sesuai dengan spesifikasi kebutuhan yang telah ditetapkan. Setelah semua fitur yang

dikembangkan selesai dan telah melalui *black box testing*, dilakukan verifikasi kepada pengguna dengan *usability testing*, dimana pengguna diminta untuk mencoba fitur-fitur website dan memberikan umpan balik terkait kemudahan penggunaan serta pengalaman mereka. *Usability testing* bertujuan untuk memastikan aplikasi tidak hanya berfungsi dengan baik tetapi juga mudah digunakan oleh target pengguna. Tahap terakhir adalah *comparative usability testing*, di mana dua versi fitur yang berbeda (versi awal dan versi perbaikan) dibandingkan untuk melihat mana yang lebih efektif.

3.3.5 Review

Tahap *review* dilakukan setelah selesai setiap *sprint*. Penulis akan mempresentasikan hasil kerja yang telah dicapai kepada tim. Dalam proses ini, penulis akan melakukan demonstrasi terhadap fitur-fitur yang dikembangkan dan tim dapat memberikan umpan balik secara langsung. Tujuan dari *review* ini adalah untuk menilai hasil *sprint*, mengidentifikasi masalah atau tantangan yang dihadapi, serta mengumpulkan informasi yang berguna untuk meningkatkan kinerja tim pada *sprint* mendatang. Jika terdapat fitur yang masih perlu diperbaiki atau belum selesai, penulis akan merencanakan untuk menyelesaikannya di *sprint* berikutnya. Evaluasi ini juga memberikan kesempatan untuk belajar dari proses yang telah dilakukan dan menerapkan perbaikan di *sprint-sprint* berikutnya.

3.3.6 Deployment

Tahap *Deployment* adalah saat dimana hasil dari *sprint-sprint* yang telah selesai dikumpulkan menjadi satu kesatuan yang siap untuk dirilis kepada pengguna. Pada tahap ini, fitur yang telah selesai dikembangkan akan dipublikasikan secara resmi dan dapat diakses pada platform. Proses *deployment* mencakup integrasi semua komponen kode pada *server* melalui halaman admin *WordPress*. Pengujian akhir dilakukan untuk memastikan bahwa semua fitur bekerja dengan baik dan siap digunakan oleh pengguna akhir.

3.3.7 *Maintenance*

Tahap *Maintenance* adalah fase yang berfokus pada pemeliharaan dan perbaikan platform setelah diluncurkan. Penulis bertanggung jawab untuk memantau pengalaman pengguna, mengelola *database*, memantau kinerja server, dan menangani masalah atau *bug* yang mungkin muncul setelah *deployment*. Selain itu, pemeliharaan meliputi pembaruan fitur dan penyesuaian berdasarkan umpan balik dari pengguna dengan mengikuti tahapan *scrum* yaitu *planning*, *development*, *testing*, dan *review* untuk memastikan efektivitas dan efisiensi. Untuk memastikan bahwa platform Quision terus berkembang sesuai dengan kebutuhan dan harapan pengguna, penulis akan terus mengamati dan menerima umpan balik secara berkala.

V. PENUTUP

5.1 Kesimpulan

Adapun kesimpulan berdasarkan penelitian yang telah dilakukan adalah sebagai berikut.

1. Pengembangan platform Quision berhasil dilaksanakan dengan metode pengembangan *Agile* dan framework *Scrum* dalam empat iterasi yang mencakup 14 *user story*, dengan fokus pada penambahan fitur filterisasi, promosi, transparansi responden, diversifikasi produk, dan optimasi fitur berdasarkan hasil evaluasi. Pelaksanaan yang telah dilakukan untuk mengembangkan platform sesuai dengan perkiraan yang telah direncanakan pada tahap perencanaan.
2. Fitur baru yang dikembangkan mendapatkan *success rate* sebesar 100%, *time-based efficiency* 0,0264, dan *error rate* 26,7% menunjukkan bahwa pengembangan fitur berhasil mencapai tujuan dalam hal keberhasilan operasional (*success rate*) yang sempurna, namun masih memiliki tantangan dan memerlukan penyempurnaan dalam hal antarmuka.
3. Hasil perbandingan antara fitur lama dan baru menunjukkan peningkatan signifikan yaitu *success rate* meningkat dari 93% (diatas rata-rata) menjadi 100% (semua pengguna berhasil menyelesaikan skenario), *time-based efficiency* meningkat hampir 3x dari 0,02 *goals*/detik menjadi 0,057 *goals*/detik, dan *error rate* turun drastis dari 90% (krusial) menjadi 3,3% (wajar).
4. Berdasarkan poin SEQ 6,4 yang diperoleh, pengembangan fitur telah memberikan kemudahan yang baik bagi pengguna, namun masukan dari pengguna menunjukkan bahwa meskipun fungsionalitas platform sudah cukup baik, masih terdapat ruang untuk perbaikan, terutama dalam hal antarmuka.

5.2 Saran

Adapun saran untuk penelitian selanjutnya adalah sebagai berikut.

1. Pengembangan lebih lanjut dapat berfokus pada penyempurnaan antarmuka pengguna berdasarkan masukan yang telah didapatkan pada hasil *usability testing*.
2. Dokumentasi tiap kode *snippet* perlu diterapkan mencakup deskripsi fungsi, lokasi implementasi, dan dependensi untuk pengembangan fitur dan *maintenance* platform yang lebih efisien.
3. Penerapan *development* dapat dilakukan pada halaman duplikat untuk meminimalisir *error* pada halaman utama. Selain itu, proses *development* dapat dilakukan pada jam dengan tingkat kunjungan pengguna yang sedikit. Hal ini menjadi salah satu solusi cepat dari kelemahan CMS *WordPress* yaitu perubahan diterapkan pada sisi *production*, sehingga pengguna dapat melihat perubahan *development* secara langsung.
4. *Repository* terpisah di luar instalasi *WordPress* dengan mengimplementasikan sistem *version control* Git dapat digunakan khusus untuk kode-kode *custom* seperti *snippet*, *functions.php*, dan modifikasi plugin untuk melacak perubahan kode dan melakukan *rollback* jika diperlukan. Hal ini menjadi solusi kelemahan CMS *WordPress* yang tidak memiliki riwayat perubahan kode.

DAFTAR PUSTAKA

DAFTAR PUSTAKA

- [1] H. R. Abubakar, *Pengantar metodologi penelitian*. SUKA-Press UIN Sunan Kalijaga, 2021.
- [2] H. F. Nasution, "Instrumen penelitian dan urgensinya dalam penelitian kuantitatif," *Al-Masharif: Jurnal Ilmu Ekonomi Dan Keislaman*, vol. 4, no. 1, hlm. 59–75, 2016, doi: 10.24952/masharif.v4i1.721.
- [3] H. Kurniawan, *Pengantar praktis penyusunan instrumen penelitian*. Deepublish, 2021.
- [4] M. B. Gainau, *Pengantar metode penelitian*. PT Kanisius, 2016.
- [5] R. S. Pressman, *Software Engineering: A Practitioner's Approach*, 7th ed. McGraw-Hill, 2010.
- [6] J. Rasmusson, "The Agile Samurai: How Agile Masters Deliver Great Software," 2010.
- [7] K. S. Rubin, *Essential Scrum: A Practical Guide to The Most Popular Agile Process*. Addison-Wesley, 2012.
- [8] F. H. Nyembe, J. A. van der Poll, dan H. H. Lotriet, "Formal Methods for an Agile Scrum Software Development Methodology," *Proceedings of the International Conference on Advanced Technologies*, vol. 11, hlm. 168–176, Agu 2023, doi: 10.58190/icat.2023.35.
- [9] J. Duckett, *Front-End Back-End Development with HTML, CSS, JavaScript, jQuery, PHP, and MySQL*, 1st ed. Wiley Publishing, 2022.
- [10] T. Arianti, A. Fa'izi, S. Adam, dan M. Wulandari, "Perancangan Sistem Informasi Perpustakaan Menggunakan Diagram UML (Unified Modelling Language)," *Jurnal Ilmiah Komputer Terapan dan Informasi*, vol. 1, no. 1, hlm. 19–25, 2022.
- [11] A. Dennis, B. Wixom, dan D. Tegarden, *Systems Analysis and Design: An Object-Oriented Approach with UML*. John Wiley & Sons, 2015.
- [12] A. Metzner, "Systematic Teaching of UML and Behavioral Diagrams," dalam *2024 36th International Conference on Software Engineering Education and Training (CSEE&T)*, IEEE, 2024, hlm. 1–5. doi: 10.48550/arXiv.2410.17849.
- [13] H. Saputra, A. Nata, dan A. P. Lubis, "Workshop Manajemen Domain Dan Hosting Website BKPRMI Kabupaten Asahan Provinsi Sumatera Utara,"

- Jurdimas (Jurnal Pengabdian Kepada Masyarakat) Royal*, vol. 6, no. 1, hlm. 152–157, Jan 2023, doi: 10.33330/jurdimas.v6i1.2018.
- [14] S. Nidhra, “Black Box and White Box Testing Techniques - A Literature Review,” *International Journal of Embedded Systems and Applications*, vol. 2, hlm. 29–50, Jun 2012, doi: 10.5121/ijesa.2012.2204.
- [15] Veni Manik, C. Hetty Primasari, Yohanes Priadi Wibisono, dan Aloysius Bagas Pradipta Irianto, “Evaluasi Usability pada Aplikasi Mobile ACC.ONE menggunakan System Usability Scale (SUS) dan Usability Testing,” *Jurnal Sains dan Informatika*, vol. 7, no. 1, hlm. 1–10, Mar 2021, doi: 10.34128/jsi.v7i1.286.
- [16] D. H. Hariri, H. Hannie, dan I. Purnamasari, “Analisis User Experience pada Website Waste4change Menggunakan Metode Single Ease Question,” *Jurnal Ilmiah Wahana Pendidikan*, vol. 8, no. 13, hlm. 95–108, 2022, doi: 10.33998/jms.2023.3.1.753.
- [17] W. Anggraeni dkk., “Implementasi CMS WordPress dalam Pengembangan website Sekolah SLB ABCD Bakti Sosial,” *Sewagati*, vol. 9, no. 1, hlm. 2639–2651, Feb 2025, doi: 10.12962/j26139960.v9i1.2321.
- [18] S. Sahiruddin, I. A. Ramadhani, dan I. Y. Pramugari, “Pengembangan Website Ikatan Keluarga Sunda Jawa Madura Kota Sorong Berbasis CMS Wordpress,” *JURNAL PETISI (Pendidikan Teknologi Informasi)*, vol. 4, no. 2, hlm. 59–68, Jul 2023, doi: 10.36232/jurnalpetisi.v4i2.4350.
- [19] C. J. Ball, *Hacking APIs: breaking web application programming interfaces*. No Starch Press, 2022.
- [20] J. Chan, *Learn SQL Using MySQL in One Day and Learn It Well: SQL for Beginners with Hands-On Project*. Packt Publishing Ltd, 2024.
- [21] D. R. Anamisa dan F. A. Mufarroha, *Dasar Pemrograman WEB Teori dan Implementasi: HTML, CSS, Javascript, Bootstrap, CodeIgniter*. Media Nusa Creative (MNC Publishing), 2022.
- [22] M. F. Santoso, “Implementation Of UI/UX Concepts And Techniques In Web Layout Design With Figma,” *Jurnal Teknologi Dan Sistem Informasi Bisnis*, vol. 6, no. 2, hlm. 279–285, Apr 2024, doi: 10.47233/jteksis.v6i2.1223.
- [23] J. Duckett, *PHP & MySQL: Server-side Web Development*. Wiley Publishing, 2022.
- [24] H. Da Rocha, *Learn Chart.js: Create interactive visualizations for the web with chart.js 2*. Packt Publishing Ltd, 2019.