

**RANCANG BANGUN *BACK END WEBSITE* QUISION:
PLATFORM PENGHUBUNG PENELITI DENGAN
RESPONDEN MENGGUNAKAN METODE *AGILE*
*DEVELOPMENT***

(Skripsi)

Oleh:

**RAIHAN FERDYANZA
2015061085**



**FAKULTAS TEKNIK
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2024**

**RANCANG BANGUN *BACK END WEBSITE* QUISION:
PLATFORM PENGHUBUNG PENELITI DENGAN
RESPONDEN MENGGUNAKAN METODE *AGILE*
*DEVELOPMENT***

Oleh

RAIHAN FERDYANZA

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

RANCANG BANGUN *BACK END WEBSITE* QUISION: PLATFORM PENGHUBUNG PENELITI DENGAN RESPONDEN MENGGUNAKAN METODE *AGILE DEVELOPMENT*

Oleh

RAIHAN FERDYANZA

Perkembangan ilmu pengetahuan sangat didorong oleh penelitian. Pengumpulan data dalam penelitian berperan krusial karena hasil dari penelitian akan menjadi landasan bagi pengembangan ilmu pengetahuan lebih lanjut. Namun dalam melakukan pengumpulan data, peneliti sering kali menghadapi tantangan, seperti responden yang tidak sesuai kriteria dan rendahnya tingkat partisipasi. Berdasarkan masalah tersebut, maka terdapat kebutuhan akan adanya platform yang mampu membantu peneliti dalam menemukan respondennya secara efisien. Penelitian ini mengembangkan platform yang bernama Quision dengan tujuan agar dapat menghubungkan peneliti dengan responden yang sesuai secara efektif dan tepat sasaran. Platform Quision dikembangkan dengan memanfaatkan metode Agile dan kerangka kerja Scrum, melalui tahapan *requirements*, *planning*, *development*, *testing*, *review*, *deployment*, dan *maintenance*. Implementasi teknologi dilakukan untuk mendukung fungsionalitas platform dalam memenuhi kebutuhan peneliti dan responden. Platform ini akan dikembangkan menggunakan *framework* WordPress yang telah disesuaikan dan diintegrasikan dengan *payment gateway*, *email protocol*, *plugins*, dan *snippet* untuk dapat memenuhi kebutuhan fitur setiap peran pengguna di platform ini. Pengujian platform ini memanfaatkan *Black Box Testing* dan *System Usability Scale* (SUS). Hasil pengujian menunjukkan bahwa platform Quision memiliki nilai usability yang dapat diterima, dengan skor SUS sebesar 77,3. Selain itu, pengujian tingkat kepuasan pengguna menunjukkan bahwa platform ini dinilai 'puas' oleh para responden. Hasil penelitian ini menunjukkan bahwa Quision berhasil dikembangkan dan mampu memfasilitasi proses pengumpulan data penelitian dengan lebih efisien.

Kata kunci: *Agile*, *Scrum*, *System Usability Scale*, *Black Box Testing*, WordPress

ABSTRACT

RANCANG BANGUN BACK END WEBSITE QUISION: PLATFORM PENGHUBUNG PENELITI DENGAN RESPONDEN MENGGUNAKAN METODE AGILE DEVELOPMENT

By

RAIHAN FERDYANZA

The advancement of knowledge is significantly driven by research. Data collection in research plays a crucial role as the outcomes serve as the foundation for further scientific development. However, researchers often face challenges in data collection, such as finding respondents who meet the criteria and low participation rates. Addressing these issues highlights the need for a platform that can efficiently assist researchers in finding suitable respondents. This research developed a platform named Quision, aimed at effectively and accurately connecting researchers with appropriate respondents. Quision was developed using Agile with Scrum methodology, through stages including requirements, planning, development, testing, review, deployment, and maintenance. Technology implementation supports the platform's functionality in meeting the needs of researchers and respondents. The platform is built using a customized WordPress framework and integrated with a payment gateway, email protocol, plugins, and snippets to fulfill the feature requirements of each user role on the platform. The platform's testing employed Black Box Testing and the System Usability Scale (SUS). The test results showed that Quision achieved an acceptable usability score of 77.3 on the SUS. Additionally, user satisfaction testing indicated that respondents rated the platform as "satisfactory." The research outcomes demonstrate that Quision was successfully developed and can facilitate a more efficient research data collection process. This platform assists researchers in obtaining valid data and provides opportunities for the community to participate in research, offering an effective solution for researchers in overcoming the challenges of questionnaire data collection.

Keywords: Agile, Scrum, System Usability Scale, Black Box Testing, WordPress.

Judul Skripsi

: **RANCANG BANGUN *BACK END*
WEBSITE QUISION: PLATFORM
PENGHUBUNG PENELITI DENGAN
RESPONDEN MENGGUNAKAN
METODE *AGILE DEVELOPMENT***

Nama Mahasiswa

: **Raihan Ferdyanza**

Nomor Pokok Mahasiswa

: **2015061085**

Program Studi

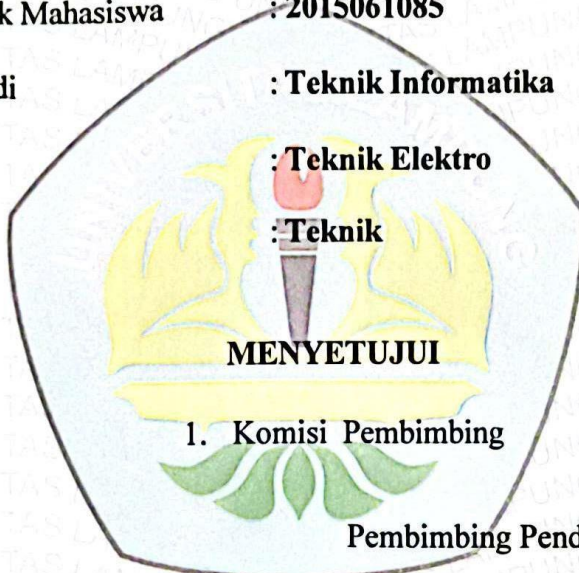
: **Teknik Informatika**

Jurusan

: **Teknik Elektro**

Fakultas

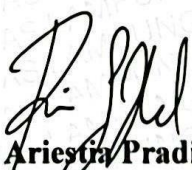
: **Teknik**



Pembimbing Utama

Pembimbing Pendamping


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


Rio Ariestia 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

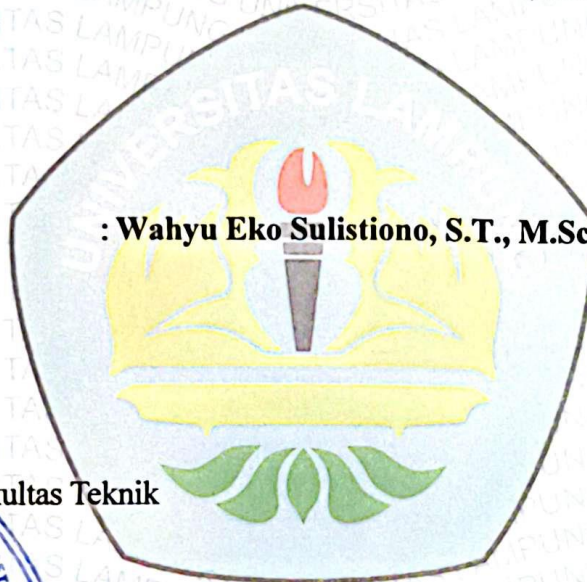
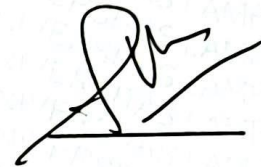
Ketua : Yessi Mulyani.,S.T.,M.T



Sekretaris : Rio Ariestia 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. J

NIP. 197509282001121002

Tanggal Lulus Ujian Skripsi : 12 Juni 2024

SURAT PERNYATAAN

Dengan ini saya menyatakan bahwa skripsi ini dengan judul “Rancang Bangun *Back End Website* Quision: Platform Penghubung Peneliti Dengan Responden Menggunakan Metode *Agile Development*” dibuat oleh saya sendiri dengan mengacu dari ide dan inovasi dari Tim PKM-K Quision dengan saya sebagai bagian dari tim. Segala hasil yang terkandung pada skripsi ini telah mengikuti kaidah penulisan karya ilmiah Universitas Lampung. Apabila di kemudian hari terbukti pernyataan saya tidak benar, maka saya bersedia dikenai sanksi sesuai dengan hukum yang berlaku.

Bandar Lampung, 2 Juli 2024
Pembuat Pernyataan,



Raihan Ferdyanza
NPM. 2015061085

RIWAYAT HIDUP



Penulis lahir di Bandar Lampung pada tanggal 28 Juni 2002 dari pasangan Bapak Drs. Sofyan Arsyad dan Ibu Zaleha. Penulis telah menyelesaikan pendidikan formal di SDN 2 Rawa Laut Bandar Lampung pada tahun 2014, SMP Kartika II-2 Bandar Lampung pada tahun 2017, dan SMAN 1 Bandar Lampung pada tahun 2020. Pada tahun 2020, penulis terdaftar sebagai mahasiswa Program Studi Teknik Informatika Universitas Lampung melalui jalur SBMPTN. Selama menjadi mahasiswa, penulis telah aktif dalam berbagai kegiatan. Menjadi anggota Himpunan Mahasiswa Teknik Elektro Universitas Lampung, terlibat dalam departemen Literasi Pengembangan periode 2020/2021. Pengembangan diri penulis tidak hanya terbatas pada lingkup kampus, penulis juga aktif mengikuti program Kampus Merdeka yaitu program Studi Independen di mitra PT. Ruang Raya Indonesia pada bidang *Front-end Engineering* dan menjalani program Magang Bersertifikat Kampus Merdeka di PT. Stechoq Robotika Indonesia sebagai *Back-End Developer* dan *Project Manager*. Selain itu, penulis berhasil mendapatkan nominasi sebagai *The Best Project Manager* selama periode magang berlangsung. Guna meningkatkan ilmu dan pengetahuan, Penulis juga aktif mengikuti pelatihan, meliputi sertifikasi BNSP untuk *Junior Web Developer* yang diadakan oleh KOMINFO, sertifikasi ASEAN *Data Scientist Explorer* yang diadakan oleh ASEAN Foundation, dan pelatihan lainnya. Tidak terbatas pada pelatihan, Penulis juga mengaplikasikan ilmu tersebut kedalam kompetisi yang diadakan oleh KEMENRISTEKDIKTI yaitu Program Kreativitas Mahasiswa Kewirausahaan dan turut andil hingga tahap puncaknya di Pekan Ilmiah Mahasiswa Nasional Ke-36, ide tersebut lah yang menjadikan landasan penulis untuk mengerjakan penelitian ini. Selain itu, Penulis juga telah melaksanakan Kuliah Kerja Nyata di Desa Tanjung Agung, Kecamatan Pakuan Ratu, Way Kanan.

MOTTO

“ Setiap kali kamu mencoba, kamu memperbesar peluangmu untuk berhasil.
Jangan biarkan ketakutan membatasi potensimu.”

- Raihan Ferdyanza

" Pengetahuan adalah jembatan antara impian dan kenyataan,
antara harapan dan pencapaian ”

- Seseorang

PERSEMBAHAN

Alhamdulillahirabbil'alamin Segala puji bagi Allah SWT. Tuhan Yang Maha Esa, karena atas segala nikmat, rahmat, dan karunia-Nya, penulis dapat menyelesaikan skripsi ini tepat pada waktunya.

Kupersembahkan Skripsi ini Kepada Orang Tuaku, Kakak, Ayuk, dan Seluruh Keluarga yang telah memberikan doa, harapan, dan semangat yang tiada hentinya sehingga penulis dapat menuntut ilmu sampai saat ini. Terima kasih dan apresiasi juga saya berikan kepada diriku sendiri selaku penulis yang telah berjuang menempuh pendidikan dan menuntut ilmu sehingga bisa membanggakan kedua orang tuanya.

SANWACANA

Alhamdulillah rabbil'alam, puji syukur penulis panjatkan kepada Allah SWT atas segala rahmat dan petunjuk-Nya yang telah diberikan, sehingga penulis berhasil menyelesaikan skripsi ini yang berjudul “Rancang Bangun *Back End Website* Quision: Platform Penghubung Peneliti Dengan Responden Menggunakan Metode *Agile Development*” tepat pada waktunya. Shalawat serta salam senantiasa juga penulis sanjungkan kepada Baginda Nabi dan Rasul Muhammad SAW, semoga syafaatnya menghampiri kita di akhirat nanti.

Proses pelaksanaan penelitian ini, melibatkan banyak pihak yang turut terlibat baik secara langsung maupun tidak langsung. Oleh karena itu, penulis mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Kedua orang tua, Papa, Mama, Kakak, dan Ayuk yang telah menjadi rumah 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 dan pembimbing utama yang telah bersedia membimbing penulis selama melaksanakan penelitian ini dan membantu proses kelancaran pengerjaan penelitian.
5. Bapak Rio Ariestia Pradipta, S.Kom, M.T.I. selaku pembimbing pendamping yang telah banyak memberikan masukan dalam proses penelitian ini.
6. Bapak Wahyu Eko Sulistiono, S.T., M.Sc. dosen penguji dan pembimbing akademik yang telah bersedia memberi banyak masukan kepada penulis selama melaksanakan penelitian dan juga saat melaksanakan proses perkuliahan.
7. Bapak Esa Ghanim Fadhallah, S.Pi., M.Si., Hafizh Zaghlul R., Afrizal Yogi P., Bella Amanda I., dan Febrina Amelia V. selaku Tim PKM-K Quision yang telah menemani, membimbing, dan turut ikut serta dalam proses pelaksanaan PKM dari pengajuan proposal hingga PIMNAS.

Penulis menyadari bahwa masih terdapat banyak kekurangan dalam penelitian ini, baik dari segi penulisan maupun pengembangan sistem. Oleh sebab itu, penulis sangat mengharapkan saran dan kritik yang membangun terkait dengan penelitian ini. Penulis juga berharap semoga penelitian ini bisa memberikan manfaat baik untuk pembaca maupun semua pihak terkait.

Bandar Lampung, 1 Maret 2024

Penulis,

A handwritten signature in black ink, appearing to read 'Raihan Ferdyanza', written in a cursive style.

Raihan Ferdyanza

DAFTAR ISI

Halaman

DAFTAR TABEL	viii
DAFTAR GAMBAR.....	ix
I. PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	3
1.3 Tujuan Penelitian.....	3
1.4 Manfaat Penelitian.....	3
1.5 Batasan Masalah	4
1.6 Sistematika Penulisan	5
II. TINJAUAN PUSTAKA.....	6
2.1 Dasar Teori	6
2.1.1 <i>Agile Software Development</i>	6
2.1.2 <i>Scrum</i>	8
2.1.3 <i>User Stories</i>	8
2.1.4 <i>Back End</i>	9
2.1.5 <i>Content Management System</i>	9
2.1.6 <i>Diagram</i>	10
2.1.7 <i>Hosting</i>	13
2.1.8 <i>Domain</i>	14
2.1.9 <i>Black Box Testing</i>	14
2.2 Penelitian Terdahulu	16
2.3 Analisis Kompetitor	18
2.4 Tools yang Digunakan	19
2.4.1 <i>WordPress</i>	19
2.4.2 <i>Plugin</i>	20
2.4.3 <i>Snippet</i>	21
2.4.4 <i>Payment gateway</i>	22
2.4.5 <i>Simple Mail Transfer Protocol (SMTP)</i>	23
2.4.6 <i>Application Programming Interface (API)</i>	24
2.4.7 <i>MySQL</i>	25
2.4.8 <i>Draw.io</i>	26
2.4.9 <i>Figma</i>	27
2.4.10 <i>PHP (Hypertext Preprocessor)</i>	27
III. METODOLOGI PENELITIAN	29

3.1	Waktu dan Tempat	29
3.1.1	Waktu.....	29
3.1.2	Tempat	30
3.2	Alat dan Bahan Penelitian.....	30
3.2.1	Alat Penelitian	30
3.2.2	Bahan Penelitian	31
3.3	Tahapan Penelitian	31
3.3.1	<i>Requirements</i>	32
3.3.2	<i>Planning</i>	33
3.3.3	<i>Development</i>	33
3.3.4	<i>Testing</i>	33
3.3.5	<i>Review</i>	34
3.3.6	<i>Deployment</i>	34
3.3.7	<i>Maintenance</i>	34
IV.	PEMBAHASAN.....	36
4.1	<i>Requirements</i>	36
4.1.1	Pengumpulan Data.....	36
4.1.2	Spesifikasi Kebutuhan	45
4.2	<i>Planning</i>	53
4.3	Iterasi 1	58
4.3.1	<i>Design</i>	58
4.3.2	<i>Implementation</i>	61
4.3.3	<i>System Testing</i>	69
4.3.4	<i>Review</i>	70
4.4	Iterasi 2	72
4.4.1	<i>Design</i>	72
4.4.2	<i>Implementation</i>	74
4.4.3	<i>System Testing</i>	82
4.4.4	<i>Review</i>	86
4.5	Iterasi 3	87
4.5.1	<i>Design</i>	87
4.5.2	<i>Implementation</i>	90
4.5.3	<i>System Testing</i>	104
4.5.4	<i>Review</i>	106
4.6	Iterasi 4	108
4.6.1	<i>Design</i>	108

4.6.2	<i>Implementation</i>	110
4.6.3	<i>System Testing</i>	118
4.6.4	<i>Review</i>	119
4.7	Iterasi 5	121
4.7.1	<i>Design</i>	121
4.7.2	<i>Implementation</i>	122
4.7.3	<i>System Testing</i>	130
4.7.4	<i>Review</i>	132
4.8	Iterasi 6	133
4.8.1	<i>Design</i>	133
4.8.2	<i>Implementation</i>	136
4.8.3	<i>System Testing</i>	144
4.8.4	<i>Review</i>	146
V.	PENUTUP	170
5.1	Kesimpulan	170
5.2	Saran	170
	DAFTAR PUSTAKA	172
	LAMPIRAN	175

DAFTAR TABEL

Tabel	Halaman
Tabel 1. Simbol dan Fungsi <i>Use Case Diagram</i>	11
Tabel 2. Simbol dan fungsi <i>Flowchart</i>	12
Tabel 3. Waktu pelaksanaan penelitian	29
Tabel 4. Perangkat keras	30
Tabel 5. Perangkat lunak	30
Tabel 6. Perbandingan Atribut WordPress, Joomla, dan Drupal	38
Tabel 7. Perbandingan SMTP Mailgun, Postmark, dan Brevo	41
Tabel 8. Data hasil kuesioner	43
Tabel 9. Analisis SWOT kompetitor	44
Tabel 10. Kebutuhan Fungsional.....	45
Tabel 11. Kebutuhan Non Fungsional.....	48
Tabel 12. <i>User Stories</i>	49
Tabel 13. Rencana Pelaksanaan Iterasi	53
Tabel 14. System <i>testing</i> iterasi ke-1	69
Tabel 15. Evaluasi estimasi dan realisasi iterasi 1	70
Tabel 16. System <i>Testing</i> Iterasi Ke-2.....	82
Tabel 17. Evaluasi Estimasi dan Realisasi Iterasi 2	86
Tabel 18. <i>System Testing</i> Iterasi Ke-3	104
Tabel 19. Evaluasi estimasi dan realisasi iterasi 3	106
Tabel 20. <i>System Testing</i> Iterasi Ke-4	118
Tabel 21. Evaluasi Estimasi dan Realisasi Iterasi 4	119
Tabel 22. <i>System Testing</i> Iterasi Ke-5	130
Tabel 23. Evaluasi estimasi dan realisasi iterasi 5	132
Tabel 24. <i>System Testing</i> Iterasi Ke-6	144
Tabel 25. Evaluasi Estimasi dan Realisasi Iterasi 6	146
Tabel 26. Daftar Identitas Responden	147
Tabel 27. Daftar Pernyataan Pengujian System Usability Scale (SUS).....	149
Tabel 28. Daftar Rekapitulasi Skor SUS Jawaban Responden	149
Tabel 29. Daftar Rekapitulasi Skor Akhir (SUS) Responden	151
Tabel 30. Data Keseluruhan Ekosistem Quision Tahun 2023-2024.....	154
Tabel 31. Daftar Pertanyaan Evaluasi Tingkat Kepuasan Pengguna	158
Tabel 31. Rekapitulasi Nilai Kuesioner Tingkat Kepuasan Pengguna.....	159
Tabel 32. Hasil Akhir Perhitungan Skor Tingkat Kepuasan Pengguna.....	161
Tabel 33. Daftar Pertanyaan Testimoni Tingkat Kepuasan Pengguna	163
Tabel 34. Daftar Jawaban Testimoni Tingkat Kepuasan Pengguna	163

DAFTAR GAMBAR

Gambar	Halaman
Gambar 1. Black Box <i>Testing</i>	14
Gambar 2. Standar penilaian SUS.....	15
Gambar 3. Tahapan <i>Scrum</i>	31
Gambar 4. Dasar Anatomi CMS	36
Gambar 5. Arsitektur Berlapis CMS	37
Gambar 6. WordPress Arsitektur.....	38
Gambar 7. <i>Use Case</i> Diagram Platform Quision	55
Gambar 8. <i>Flowchart</i> Diagram Platform Quision	57
Gambar 9. Desain Halaman Beranda	58
Gambar 10. Desain Halaman Masuk ke Akun	58
Gambar 11. Desain Halaman Mendaftar Akun	59
Gambar 12. Desain Informasi Akun dan Dropdown untuk Keluar dari Akun.....	59
Gambar 13. Desain <i>Edit</i> Informasi Akun.....	60
Gambar 14. Desain Halaman untuk <i>Reset Password</i> Akun	60
Gambar 15. Tampilan Halaman Beranda	61
Gambar 16. Tampilan Halaman Masuk ke Akun	61
Gambar 17. Tampilan Halaman Registrasi Akun.....	62
Gambar 18. Tampilan Halaman Profil Akun.....	63
Gambar 19. Tampilan Halaman <i>Edit</i> Profil Akun	63
Gambar 20. Tampilan Halaman <i>Reset Password</i>	64
Gambar 21. Pembuatan <i>Field</i> Akun	65
Gambar 22. Mengonfigurasi OAuth <i>Credentials</i> dengan Google.....	66
Gambar 23. Konfigurasi dan Pembuatan <i>Email Domain</i>	66
Gambar 24. Mengonfigurasi <i>E-Mail Domain</i> dengan Brevo	67
Gambar 25. Kode Perubahan <i>Password</i> pada Ultimate Member.....	68
Gambar 26. Desain Halaman Kumpulan Kuesioner	72
Gambar 27. Desain Halaman Detail Kuesioner	72
Gambar 28. Desain Halaman Penerbitan Kuesioner.....	73
Gambar 29. Desain Halaman <i>Form</i> Permintaan Perubahan Kuesioner	73
Gambar 30. Desain Halaman Detail Kuesioner untuk Admin	74
Gambar 31. Tampilan Halaman Kumpulan Kuesioner	74
Gambar 32. Tampilan Halaman Detail Kuesioner	75
Gambar 33. Tampilan Halaman Penerbitan Kuesioner.....	76
Gambar 34. Tampilan Halaman Revisi Kuesioner	76
Gambar 35. Halaman Akses Kuesioner untuk Admin	77
Gambar 36. Konfigurasi <i>Field Form</i> Penerbitan Kuesioner.....	78
Gambar 37. Kode Pembatasan Akses Tautan <i>Form</i> Kuesioner	79
Gambar 38. Tampilan Section Strict Tautan <i>Form</i> Kuesioner	79
Gambar 39. Kode Perhitungan Total Responden yang Telah Mengisi Kuesioner	80
Gambar 40. Tampilan Total Responden yang Telah Mengisi Kuesioner.....	80
Gambar 41. Konfigurasi <i>Form</i> untuk Revisi Kuesioner	81

Gambar 42. Mengisi Kuesioner Sesuai Kriteria	87
Gambar 43. Mengisi Kuesioner Tidak Sesuai Kriteria	87
Gambar 44. Mengisi <i>Form Reward</i>	88
Gambar 45. Mengakses Pengisian Kuesioner	88
Gambar 46. Melihat Engagement Kuesioner	89
Gambar 47. Desain Halaman Riwayat Status Pengisian Kuesioner	89
Gambar 48. Tampilan Detail Kuesioner jika Responden Sesuai Kriteria	90
Gambar 49. Tampilan Detail Kuesioner jika Responden Tidak Sesuai Kriteria...	91
Gambar 50. Tampilan Responden Telah Mengisi Kuesioner	92
Gambar 51. Tampilan <i>Form</i> Konfirmasi Pengisian	93
Gambar 52. Tampilan Peninjauan Pengisian Kuesioner untuk Admin	94
Gambar 53. Tampilan Halaman Riwayat Status Pengisian Kuesioner	95
Gambar 54. Tampilan <i>Dashboard</i> Kuesioner.....	96
Gambar 55. Kode Konfigurasi Akses Pengisian Kuesioner.....	97
Gambar 56. Kode Konfigurasi Data Diri Belum Lengkap	99
Gambar 57. Kode Konfigurasi Validasi Kriteria Responden	100
Gambar 58. Konfigurasi <i>Form</i> Konfirmasi Pengisian Kuesioner.....	102
Gambar 59. Kode Konfigurasi <i>Chart Dashboard</i> Pengguna	103
Gambar 60. Desain halaman Log Qu Coin	108
Gambar 61. Desain halaman akses Qu Coin Pengguna	108
Gambar 62. Desain halaman pencairan Qu Coin	109
Gambar 63. Desain Halaman Riwayat Pencairan Qu Coin	109
Gambar 64. Desain Halaman Peninjauan Pencairan Qu Coin	110
Gambar 65. Tampilan Halaman Pencairan Qu Coin	110
Gambar 66. Tampilan Halaman Peninjauan Pencairan Qu Coin	111
Gambar 67. Tampilan Halaman Log Qu Coin	112
Gambar 68. Tampilan Halaman Riwayat Pencairan Qu Coin.....	113
Gambar 69. Tampilan Halaman Akses Qu Coin Pengguna	113
Gambar 70. Kode Transaksi Qu Coin dengan Integrasi WooCommerce.....	114
Gambar 71. Kode Log Riwayat Qu Coin dengan Integrasi WooCommerce	115
Gambar 72. Kode Log Riwayat Qu Coin dengan Integrasi WooCommerce	116
Gambar 73. Desain Halaman Pemesanan Responden.....	121
Gambar 74. Desain Halaman Pembayaran Pesanan Responden.....	121
Gambar 75. Desain Halaman Riwayat Pembayaran Pesanan	122
Gambar 76. Tampilan Halaman Pesan Responden	122
Gambar 77. Tampilan Halaman Pembayaran Pesanan Responden.....	123
Gambar 78. Tampilan Halaman Riwayat Pembayaran Pesanan	124
Gambar 79. Konfigurasi Produk Jasa Responden Penelitian.....	124
Gambar 80. Konfigurasi <i>Email</i> Pesanan Produk	125
Gambar 81. Aktivasi Akun Payment Gateway Xendit	125
Gambar 82. Konfigurasi dan Integrasi API <i>Payment gateway</i> Xendit.....	126
Gambar 83. Aktivasi Metode Pembayaran Melalui <i>E-Wallets</i> dan QRIS.....	127
Gambar 84. Integrasi Metode Pembayaran Melalui <i>E-Wallets</i> dan QRIS	127
Gambar 85. Perubahan <i>Role</i> Menjadi Peneliti Setelah Melakukan Pembayaran	128
Gambar 86. Perubahan <i>Role</i> Menjadi Responden Setelah Penerbitan.....	129

Gambar 87. Desain Halaman Peneliti Meminta Pengembalian Dana	133
Gambar 88. Desain Halaman Admin Meninjau Pengembalian Dana	133
Gambar 89. Desain Halaman Panduan Alur Responden.....	134
Gambar 90. Desain Halaman Panduan Alur Peneliti	134
Gambar 91. Desain <i>Form</i> Saran dan Informasi Kontak.....	135
Gambar 92. Tampilan <i>Form</i> Pengembalian Dana untuk Peneliti.....	136
Gambar 93. Tampilan Halaman Admin Permintaan Pengembalian Dana	137
Gambar 94. Tampilan Halaman Panduan Responden	138
Gambar 95. Tampilan Halaman Panduan Peneliti	139
Gambar 96. Tampilan Halaman Informasi Kontak	140
Gambar 97. Tampilan <i>Form</i> Saran dan Masukan	141
Gambar 98. <i>Form</i> Pengajuan Permohonan Pengembalian Dana untuk Peneliti.	142
Gambar 99. Konfigurasi <i>Form</i> Saran dan Masukan	143
Gambar 100. Data Jumlah Pengguna Platform Qusion	154
Gambar 101. Data Jumlah Responden Dipesan pada Platform Qusion.....	155
Gambar 102. Data Jumlah Peneliti Melakukan Transaksi Platform Qusion	156
Gambar 103. Data Jumlah Keuntungan pada Platform Qusion	157

I. PENDAHULUAN

1.1 Latar Belakang

Perkembangan ilmu pengetahuan tidak lepas dari banyaknya penelitian yang telah dilakukan hingga saat ini sehingga memiliki peran yang penting dalam memberikan berbagai pengetahuan baru untuk mengatasi beragam tantangan dan permasalahan [1]. Oleh karena itu, penelitian harus berjalan dengan baik terutama dalam proses pengumpulan data untuk dapat memastikan hasil data yang akurat dan implementasinya dapat berjalan dengan lancar sehingga bisa menjadi landasan untuk pengembangan ilmu pengetahuan di masa depan. Pengumpulan data dalam penelitian dapat menggunakan berbagai metode penelitian seperti diskusi, eksperimen, observasi, *interview*, dan kuesioner [2].

Salah satu metode yang sering dipakai dalam menghimpun informasi adalah kuesioner. Kuesioner umumnya dipilih karena mengandung serangkaian pertanyaan yang beragam untuk semua responden [3]. Model kuesioner yang sedang populer saat ini adalah kuesioner yang dibuat berbasis *website*. Kuesioner *online* ini dipilih karena kemampuannya untuk mengumpulkan data tanpa terbatas oleh ruang dan waktu.

Di masa kini, terdapat berbagai platform pembuat kuesioner seperti Google Form, Type Form dan platform serupa lainnya. Namun, platform-platform tersebut hanya menyediakan alat untuk membuat kuesioner, sedangkan pengguna harus menyebarluaskan kuesioner secara independen. Dalam proses tersebut, timbul beberapa hambatan, seperti responden yang tidak memenuhi kriteria penelitian dan Tingkat partisipasi yang rendah [4]. Hal ini mengakibatkan kurangnya validitas data karena kesesuaian yang kurang antara responden dan waktu survei, sehingga efektivitas survei menjadi berkurang.

Berdasarkan masalah yang telah diuraikan sebelumnya, terdapat kebutuhan akan adanya platform yang mampu menghubungkan peneliti dengan responden secara efektif dan efisien. Meskipun telah dilakukan analisis pasar, masih jarang ditemukan platform yang menyediakan layanan dengan cepat,

kriteria responden yang tepat, dan data yang akurat. Oleh karena itu, Quision hadir untuk menyelesaikan masalah tersebut. Quision merupakan sebuah platform digital yang bertujuan untuk memfasilitasi koneksi antara peneliti dan responden dengan cepat, akurat, dan tepat sasaran. Quision melibatkan masyarakat langsung sebagai responden yang terdaftar di dalamnya. Diharapkan peneliti dapat memperoleh responden dengan cepat dan efisien, serta mempercepat proses pengembangan ilmu pengetahuan dengan adanya Quision.

Dibutuhkan sistem *back end* yang handal dan fleksibel serta mampu mengelola data responden, memfasilitasi proses pemesanan responden, dan menyediakan layanan atau fitur yang sesuai dengan kebutuhan responden maupun peneliti untuk mencapai tujuan yang diharapkan di platform Quision. *Back end* pada platform Quision berperan besar dalam pengelolaan sistem dan *database* agar data dan tampilan pada platform dapat saling berkomunikasi dengan baik guna mendukung antarmuka sistem bekerja sesuai dengan fungsinya [5]. Pengembangan *back end website* Quision perlu dilakukan dengan pendekatan yang adaptif dan responsif terhadap perubahan kebutuhan pengguna.

Pengembangan *back end website* Quision dapat dilakukan secara fleksibel dengan menerapkan metode *Agile Development*, memungkinkan penyesuaian dengan kebutuhan dan perubahan yang mungkin terjadi selama proses pengembangan. Keunggulan lainnya dari metode *Agile* adalah dari proses pengembangannya yang memungkinkan kolaborasi serta mengevaluasi satu sama lain antar tim baik dari sisi *back end* maupun *front end* dengan waktu pengembangan yang singkat [6]. Dengan demikian, penggunaan metode *Agile Development* dalam rancang bangun *back end website* Quision tidak hanya akan memastikan perancangan dibuat sesuai dengan kebutuhan pengguna dan dapat diselesaikan tepat waktu, tetapi juga akan memungkinkan platform Quision untuk terus berkembang dan meningkatkan layanannya dalam jangka waktu yang singkat.

1.2 Rumusan Masalah

1. Bagaimana cara mengembangkan platform yang dapat memfasilitasi responden yang sesuai dengan kriteria peneliti dari segi *back end*?
2. Bagaimana cara menerapkan teknologi yang tepat sesuai kebutuhan pengguna dengan waktu pengembangan yang relatif singkat?
3. Bagaimana platform yang telah dibuat telah mencapai tingkat kegunaan atau *usability* yang dapat diterima oleh pengguna?

1.3 Tujuan Penelitian

Adapun tujuan dari penelitian ini di antaranya adalah:

1. Mengembangkan platform Quision dengan metode *Agile* dan kerangka kerja *Scrum* yang efisien dan efektif.
2. Mengimplementasikan teknologi *back end* pada seluruh fitur di platform Quision dalam memfasilitasi responden sesuai dengan kriteria peneliti.
3. Menguji fungsionalitas pada platform Quision agar sesuai dengan kebutuhan pengguna.

1.4 Manfaat Penelitian

Adapun manfaat dari penelitian ini di antaranya adalah:

1. Penelitian ini akan mendukung terciptanya sebuah platform yang memfasilitasi pengumpulan data responden dengan lebih efisien, membantu peneliti mendapatkan informasi yang diperlukan dengan lebih cepat dan tepat.
2. Bagi peneliti, peneliti akan memiliki akses yang lebih mudah untuk mencari dan mendapatkan responden yang sesuai dengan kriteria penelitian mereka, membantu dalam kelancaran dan efisiensi proses penelitian.
3. Bagi responden, responden akan memiliki akses yang lebih mudah dan fleksibel untuk berpartisipasi dalam penelitian yang relevan dengan minat atau latar belakang mereka.
4. Penelitian ini akan membantu memfasilitasi berbagai penelitian dengan metode kuesioner agar lebih efektif dan efisien, pengetahuan yang dihasilkan dari

penelitian tersebut dapat digunakan untuk memecahkan berbagai permasalahan sosial, ekonomi, dan lingkungan, yang pada akhirnya akan meningkatkan kesejahteraan masyarakat.

5. Hasil platform yang dibuat memiliki peluang usaha untuk ditawarkan kepada peneliti yang sedang membutuhkan responden sehingga hasil dari penelitian memiliki nilai ekonomi yang terkandung di dalamnya.
6. Penelitian ini dapat menjadi sumber referensi dan pijakan bagi penelitian lanjutan di bidang yang relevan.

1.5 Batasan Masalah

Adapun batasan masalah pada penelitian ini di antaranya adalah:

1. Penelitian ini termasuk ke dalam Program Kreativitas Mahasiswa Kewirausahaan (PKM-K) dan telah didanai oleh Dikti (Direktorat Jenderal Pendidikan Tinggi, Riset, dan Teknologi) sehingga pada penelitian ini hanya akan membahas pembuatan platform hingga tahap platform beroperasi dan dapat dipasarkan, sesuai dengan tenggat waktu yang diberikan yaitu tiga bulan setelah pendanaan.
2. Penelitian ini hanya memanfaatkan WordPress sebagai *open-source Content Management System* (CMS) yang telah termodifikasi dan terintegrasi dengan *payment gateway*, *Simple Mail Transfer Protocol* (SMTP), *plugins*, dan *snippet* untuk menyesuaikan kebutuhan fitur pengguna platform Quision.
3. Penelitian ini dikembangkan dengan kolaborasi antara anggota tim PKM-K, yaitu penulis yang mengembangkan dari aspek *backend* serta anggota lainnya mengembangkan dari aspek *frontend*, *maintanance*, dan *improvement* sehingga pada penelitian ini lebih banyak berfokus pada pengembangan aplikasi dari sisi *backend*.

1.6 Sistematika Penulisan

Adapun struktur penulisan yang diterapkan pada penelitian ini sebagai berikut:

BAB I: PENDAHULUAN

Menjelaskan latar belakang, rumusan masalah, tujuan penelitian, dan batasan masalah dari pengembangan platform Quision, serta menuliskan susunan sistematika laporan.

BAB II: TINJAUAN PUSTAKA

Meninjau penelitian terdahulu dengan alat, cara kerja, dan metode pengembangan yang relevan dan sesuai dalam pengembangan platform ini serta dasar-dasar teoritis yang didasarkan pada literatur dari berbagai sumber seperti buku, jurnal, skripsi, dan analisa kompetitor.

BAB III: METODE PENELITIAN

Menguraikan langkah-langkah sistematis dan logis yang memuat waktu, tempat, alat, bahan, dan tahapan penelitian dalam pembuatan dan pengembangan platform Quision dengan menggunakan metode *Agile* kerangka kerja *Scrum*.

BAB IV: HASIL DAN PEMBAHASAN

Menyajikan desain, hasil implementasi, pengujian, dan evaluasi untuk tiap iterasi yang dikerjakan dengan mengacu pada metode *Agile* kerangka kerja *Scrum*.

BAB V: KESIMPULAN DAN SARAN

Berisi kesimpulan dari hasil pembuatan dan pengembangan platform Quision serta saran untuk pengembangan penelitian selanjutnya.

DAFTAR PUSTAKA

LAMPIRAN

II. TINJAUAN PUSTAKA

2.1 Dasar Teori

2.1.1 *Agile Software Development*

Agile Software Development merupakan salah satu metodologi pengembangan perangkat lunak yang menekankan pada kerjasama tim, adaptasi terhadap perubahan yang cepat, dan pengembangan perangkat lunak dengan iterasi pendek. Metode ini menekankan pada siklus pengembangan yang berulang-ulang (iteratif) dan inkremental, di mana fitur-fitur baru atau perbaikan diperkenalkan secara bertahap dalam setiap iterasi [7]. *Agile* sangat cocok untuk proyek-proyek di mana kebutuhan mungkin berubah selama proses pengembangan, atau di mana pentingnya mendapatkan produk yang berfungsi dengan cepat lebih diutamakan daripada memiliki produk yang sepenuhnya matang. Metode pengembangan ini telah menjadi sangat populer di industri perangkat lunak karena fleksibilitasnya dan kemampuannya untuk memberikan nilai tambah yang cepat kepada pelanggan. Beberapa kerangka kerja yang umum dalam pengembangan perangkat lunak *Agile* termasuk *Scrum*, *Kanban*, dan *Extreme Programming (XP)*.

Metode *Agile* melibatkan serangkaian tahapan secara umum, termasuk perencanaan, implementasi, *testing*, dokumentasi, *deployment*, dan pemeliharaan [8]. Pada implementasi pengerjaannya metode ini memiliki tahapan kerja yang bersifat tidak kaku dan tergantung dari kerangka kerja yang digunakan. Kerangka kerja pada metode *Agile* umumnya memiliki perencanaan, pembuatan *user stories*, pembagian dan pengerjaan tugas, pengembangan dan pengujian, integrasi, *sprint review*, dan perubahan atau pembaruan *backlog*. Perencanaan dimulai dengan tim pengembang maupun divisi lain merencanakan iterasi berikutnya dengan memilih dan memprioritaskan *item backlog* (daftar pekerjaan yang harus dilakukan) untuk dipecah menjadi tugas-tugas yang dapat diselesaikan dalam iterasi tersebut. *User stories* adalah deskripsi singkat dari fitur atau fungsionalitas yang diinginkan oleh pengguna dan membantu dalam pemahaman kebutuhan pengguna untuk menjadi landasan dalam pengembangan selanjutnya. Tim pengembang membagi *User*

stories menjadi tugas-tugas yang lebih kecil yang dapat dikerjakan dalam iterasi tersebut. Tugas-tugas ini biasanya memuat pembuatan kode dan pengujian menguji fitur yang dikembangkan serta diberikan estimasi waktu untuk menyelesaikannya.

Setiap perubahan kode yang dibuat oleh pengembang diintegrasikan secara terus-menerus ke dalam sistem utama dan tetap memastikan bahwa perubahan baru tidak mengganggu pekerjaan tim yang lain. Setelah tugas-tugas selesai dilakukan, tim pengembang melakukan demonstrasi tentang apa yang telah dicapai. Umpan balik dari demonstrasi ini digunakan untuk memperbaiki dan meningkatkan produk. Setelah iterasi selesai, tim melaksanakan *sprint review*, di mana mereka mengevaluasi proses pengembangan mereka. Mereka membahas apa yang telah berhasil dilakukan, apa yang bisa diperbaiki, dan membuat rencana untuk perbaikan di masa depan. Berdasarkan umpan balik dari *sprint review* dan demonstrasi, item *Backlog* diperbarui atau disesuaikan untuk iterasi berikutnya. Hal ini memastikan bahwa prioritas yang tepat diberikan untuk fitur-fitur yang paling penting. Setelah iterasi selesai, siklus ini berulang kembali dengan tim melakukan iterasi berikutnya

Metode ini menempatkan fokus utama pada kebutuhan *User* yang berkelanjutan. Berbeda dengan pendekatan tradisional yang mengharuskan perencanaan rinci di awal dan menerapkan perubahan dengan sulit di tengah jalan, *Agile* menekankan fleksibilitas, adaptasi, dan pengiriman hasil yang bernilai secara terus-menerus. Hal ini memungkinkan adaptasi yang cepat terhadap perubahan kebutuhan, sehingga menghasilkan produk yang lebih tepat dan efisien [9]. Tim yang menggunakan pendekatan *Agile* biasanya mengadopsi siklus pengembangan iteratif dan inkremental, di mana pekerjaan direncanakan dalam *sprint* atau iterasi pendek, dan umpan balik diperoleh dari pelanggan secara teratur untuk memandu pengembangan selanjutnya. Metodologi ini juga berdampak positif pada biaya pengembangan, waktu, dan produktivitas, yang mengarah pada meningkatnya keberhasilan proyek perangkat lunak. Dengan demikian, *Agile* memungkinkan tim untuk memberikan nilai tambah yang cepat kepada pelanggan sambil tetap fleksibel dan responsif terhadap perubahan yang terjadi.

2.1.2 *Scrum*

Scrum adalah salah satu kerangka kerja pengembangan perangkat lunak yang termasuk dalam metodologi *Agile* yang paling populer dan banyak digunakan dalam pengembangan produk secara iteratif dan inkremental. Dalam kerangka *Scrum*, proyek dikelompokkan ke dalam iterasi pendek yang disebut *Sprint*, biasanya berlangsung selama satu hingga empat minggu. Setiap *Sprint* dimulai dengan pertemuan perencanaan *Sprint*, diikuti oleh serangkaian kegiatan pengembangan, termasuk pertemuan per iterasi, demo fitur, dan *sprint review*. *Scrum* memberikan fokus pada integrasi pengembangan, kolaborasi tim, dan dapat menangani secara cepat perubahan yang terjadi pada proses pengembangan [10]. Tim pengembang dapat mempercepat *product delivery*, meningkatkan kualitas, dan merespons perubahan kebutuhan menjadi lebih efektif dengan menerapkan kerangka kerja *Scrum*.

Kerangka kerja pengembangan perangkat lunak *Scrum* menekankan pada iterasi yang cepat. Tahapan implementasi *Scrum* dimulai pada tahapan perencanaan dengan membuat *product backlog*, yang merupakan daftar kebutuhan dan fitur produk. Setiap *sprint* dimulai dengan pertemuan perencanaan *Sprint*, di mana tim memilih daftar pekerjaan dari *product backlog* untuk diimplementasikan dalam *sprint* tersebut [11]. Selama *sprint*, tim bekerja secara kolaboratif untuk mengembangkan fungsionalitas yang dipilih. Setiap beberapa minggu, tim melakukan pertemuan untuk memberikan laporan progres, identifikasi hambatan, dan memastikan keterlibatan seluruh anggota tim. Setelah selesai *Sprint*, dilakukan pertemuan *Review sprint* untuk memperlihatkan hasil kerja kepada pemangku kepentingan dan mendapatkan umpan balik. Terakhir, tim melakukan pertemuan *sprint review* untuk mengevaluasi proses dan menentukan perbaikan untuk *Sprint* berikutnya. Dengan mengikuti tahapan ini secara teratur, tim dapat memaksimalkan efisiensi pengembangan, meningkatkan kualitas produk, dan merespons perubahan kebutuhan dengan lebih cepat dan fleksibel.

2.1.3 *User Stories*

User stories adalah teknik pengumpulan kebutuhan dalam pengembangan perangkat lunak yang berfokus pada pengalaman pengguna. *User stories* berisi

nama pengguna sistem dan fitur-fitur yang menjadi kebutuhan sistem dengan bahasa pengguna secara umum [10]. Dalam metode *Scrum*, *User stories* digunakan untuk mewakili kebutuhan fungsionalitas atau fitur yang diinginkan oleh pengguna akhir. Setiap *User story* biasanya terdiri dari tiga elemen: 'Siapa', 'Apa', dan 'Mengapa'. 'Siapa' mengidentifikasi pengguna yang terlibat, 'Apa' menjelaskan kebutuhan atau fitur yang diinginkan, dan 'Mengapa' menjelaskan tujuan atau manfaat yang ingin dicapai. *User stories* digunakan sebagai alat untuk memprioritaskan dan mengorganisir pekerjaan dalam *Sprint*, dengan memastikan bahwa pengembangan fungsionalitas dilakukan dengan fokus pada nilai bisnis yang sebenarnya bagi pengguna. Melalui penggunaan *User stories*, tim *Scrum* dapat memahami kebutuhan pengguna secara lebih baik, memungkinkan untuk menghasilkan produk yang lebih relevan dan memuaskan bagi pengguna akhir.

2.1.4 Back End

Back-End dalam konteks pengembangan perangkat lunak merujuk pada bagian dari sistem yang bertanggung jawab untuk memproses data, menjalankan logika bisnis, dan berinteraksi dengan *database*. *Backend* merupakan inti dari aplikasi *web* atau perangkat lunak, dan sering kali tidak terlihat oleh pengguna akhir. Itu berfungsi untuk mengelola segala sesuatu yang terjadi di belakang layar, seperti pengolahan permintaan pengguna, pengaturan keamanan, dan menyimpan data yang diperlukan. Selain itu, *backend* juga mendukung pemrosesan data dalam hal *parsing*, *validasi*, *routing*, dan *request* [12]. Aplikasi dapat berjalan secara lancar, aman, dan dapat diandalkan, dengan *backend* yang solid dan efisien serta memberikan pengalaman pengguna yang lebih baik secara keseluruhan.

2.1.5 Content Management System

Content Management System atau CMS adalah sistem yang memiliki fasilitas untuk dapat mengolah, mengatur dan memanfaatkan data dengan antarmuka yang intuitif dan responsif. CMS dirancang untuk *developer* dapat melakukan manajemen konten, *templating*, manajemen pengguna, manajemen hak akses, dengan cepat dan efisien. CMS yang beredar saat ini umumnya dibangun dan dikembangkan untuk fleksibilitas dan membagi kerangka kerjanya menjadi *single*

modular packages, yang membantu setiap *package* dirancang lebih efisien [13]. Beberapa contoh CMS yang populer termasuk WordPress, Drupal, Joomla, dan Shopify.

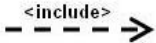
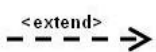
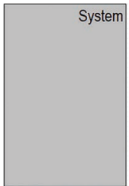
2.1.6 Diagram

1. *Use Case Diagram*

Use Case Diagram merupakan pemodelan yang menggambarkan berbagai skenario penggunaan atau kelakuan (*behavior*) yang mungkin terjadi dalam suatu sistem [14]. Fokus utama dari diagram ini adalah menjelaskan bagaimana aktor satu dengan lainnya berinteraksi dengan sistem. Sebagai konsep fundamental dalam pengembangan sistem, diagram *Use Case* digunakan pada tahap analisis dan desain sistem untuk memodelkan fungsionalitas yang diharapkan. Terdapat komponen-komponen yang ada pada *Use Case Diagram* seperti *Use Cases*, aktor, asosiasi, dan relasi antar *Use Cases*, diagram ini mampu menggambarkan siapa yang berinteraksi dengan sistem, apa yang dapat dilakukan oleh pengguna, serta bagaimana hubungan antar berbagai fungsionalitas dalam sistem.

Setiap elemen dalam diagram *Use Case* memiliki peran dan representasi tersendiri. *Use Cases* menggambarkan fungsionalitas yang diinginkan dari sistem yang dikembangkan, sementara aktor mewakili pengguna atau entitas eksternal yang berinteraksi dengan sistem. Asosiasi digunakan untuk menghubungkan aktor dengan *Use Cases*, sementara relasi antar *Use Cases* menunjukkan hubungan antara berbagai fungsionalitas dalam sistem. Terdapat dua jenis relasi antar *Use Cases*, yaitu *include* dan *extend*, yang masing-masing mengindikasikan ketergantungan dan perluasan antara fungsionalitas yang dijelaskan dalam diagram [15]. Dengan demikian, diagram *Use Case* memberikan gambaran yang jelas tentang bagaimana pengguna akan menggunakan sistem serta bagaimana fungsionalitas sistem saling berinteraksi satu sama lain.

Tabel 1. Simbol dan Fungsi *Use Case Diagram*

NO	Simbol	Nama	Kegunaan
1		Aktor	Pengguna yang berinteraksi dengan sistem. Aktor bisa berupa individu, organisasi, atau sistem eksternal..
2		<i>Use Case</i>	Proses Bisnis dan interaksi antara sistem dan aktor.
3		<i>Association</i>	Relasi satu arah di mana seorang aktor bertanggung jawab atas pengaruh terhadap sebuah <i>Use Case</i> .
4		<i>Generalization</i>	Relasi antara sebuah <i>Use Case</i> induk dan satu atau lebih <i>Use Case</i> anak.
5		<i>Include</i>	Relasi di mana satu <i>Use Case</i> mencakup fungsionalitas dari satu <i>Use Case</i> lainnya.
6		<i>Extend</i>	Relasi di mana <i>Use Case</i> yang diperluas menambahkan perilaku (<i>behaviour</i>) tambahan terhadap fungsionalitas yang sudah ada dari <i>Use Case</i> dasar.
7		<i>System Boundary</i>	Mewakili cakupan sistem. Menyatukan seluruh fungsionalitas sistem.

2. Flow Chart

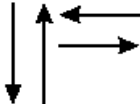
Flow chart merupakan pemodelan yang menggambarkan aliran kerja dalam suatu proses bisnis atau sistem secara jelas, ringkas, dan logis dalam bentuk diagram [16]. *Flowchart* juga membantu para pengembang perangkat lunak untuk memahami, merancang, dan memvisualisasikan proses bisnis atau alur kerja dalam suatu sistem dengan jelas dan terstruktur. Pada *Flowchart*, semua alur didokumentasikan dengan langkah-langkah teknis yang diperlukan untuk

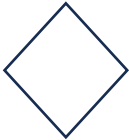
menyelesaikan sebuah tugas atau mencapai sebuah tujuan. *Flowchart* secara teknis sering digunakan oleh para programmer untuk merencanakan algoritma atau struktur program.

Pada praktiknya, pembuatan *Flowchart* melibatkan pemilihan simbol-simbol standar yang mewakili berbagai jenis instruksi atau kegiatan, seperti pengambilan keputusan, proses, *input* dan *output*, dan lain-lain. Setiap simbol ini dihubungkan dengan panah yang menunjukkan aliran atau urutan langkah-langkah. *Flowchart* membantu dalam visualisasi proses, komunikasi, analisis, perencanaan, dan pemrograman, sehingga menjadi alat yang sangat berguna dalam berbagai bidang seperti pemrograman komputer, rekayasa perangkat lunak, manajemen proyek, dan proses bisnis.

Tabel 2 di bawah ini merupakan beberapa simbol *Flowchart* yang paling sering digunakan, khususnya dalam penelitian ini.

Tabel 2. Simbol dan fungsi *Flowchart*

NO	Simbol	Nama	Kegunaan
1		<i>Flow Direction</i>	Menggambarkan arah aliran dari satu langkah atau tindakan ke langkah atau tindakan berikutnya dalam <i>Flowchart</i> . Biasanya berbentuk panah dengan ujung di bagian bawah.
2		<i>Terminator</i>	Menandai awal atau akhir dari algoritma atau proses dalam <i>Flowchart</i> dan menunjukkan titik awal atau akhir dari alur proses, biasanya berbentuk persegi panjang dengan sudut yang dibulatkan.
3		<i>Processing</i>	Mewakili langkah-langkah atau tindakan yang dilakukan dalam proses. Ini bisa menjadi langkah-

			langkah perhitungan, manipulasi data, atau tindakan lainnya yang diperlukan dalam algoritma.
4		<i>Decision</i>	Menunjukkan pengambilan keputusan dalam alur proses, berbentuk <i>diamond</i> dan memiliki dua cabang keluar, mewakili hasil dari keputusan (ya atau tidak).
5		<i>Page Connector</i>	Simbol ini digunakan untuk menghubungkan bagian-bagian dari <i>Flowchart</i> yang terpisah karena halaman yang berbeda,
6		<i>Input-Output</i>	Mewakili operasi masukan (<i>input</i>) atau keluaran (<i>output</i>) dalam proses, dapat berupa pembacaan data dari pengguna atau perangkat, atau penulisan hasil atau informasi ke layar

2.1.7 *Hosting*

Hosting adalah layanan yang menyediakan infrastruktur dan teknologi yang diperlukan untuk menyimpan dan mengakses situs *web*, aplikasi, atau data secara *online* sehingga bisa diakses oleh orang lain [17]. Melalui *hosting*, pengguna dapat menyimpan *file-file* yang diperlukan untuk situs *web* mereka, seperti kode HTML, gambar, video, dan data basis data, pada *server* yang terhubung ke internet. *Hosting* juga menyediakan sumber daya komputasi dan jaringan yang diperlukan untuk memastikan situs *web* dapat diakses oleh pengguna dari seluruh dunia dengan cepat dan andal. Terdapat berbagai jenis *hosting*, termasuk *shared hosting*, VPS (*Virtual Private Server*) *hosting*, dan *cloud hosting*, masing-masing dengan kelebihan dan kekurangannya sendiri. Dengan menggunakan layanan *hosting*, suatu organisasi atau perusahaan dapat membuat dan mengelola situs *web* mereka secara *online*

tanpa perlu memiliki infrastruktur fisik sendiri, menjadikannya kunci dalam membangun kehadiran digital yang kuat dan sukses.

2.1.8 Domain

Domain adalah sebuah alamat unik yang digunakan sebagai pengenalan untuk mengidentifikasi dan mengakses situs *web* atau layanan *online* tertentu di internet [17]. *Domain* biasanya terdiri dari nama yang mudah diingat diikuti oleh ekstensi seperti .com, .net, .org, dan sebagainya. Setiap *domain* harus terdaftar secara resmi dan unik di sistem *domain name system* (DNS) yang dikelola oleh organisasi terkait. *Domain* memberikan identitas dan branding yang mudah diingat untuk situs *web* atau bisnis *online*, serta memungkinkan pengguna untuk mengaksesnya dengan mudah melalui peramban *web* mereka. Selain itu, *domain* juga dapat digunakan untuk membuat alamat *email* khusus yang terkait dengan situs *web* atau bisnis tersebut. Dengan memiliki *domain* yang tepat dan mudah diingat, individu dan bisnis dapat meningkatkan kehadiran mereka secara *online* dan memperkuat citra merek mereka di internet.

2.1.9 Black Box Testing

Black Box Testing adalah metode pengujian perangkat lunak di mana pengujian dilakukan tanpa mengetahui atau memperhatikan struktur internal atau kode sumber dari perangkat lunak yang diuji [18]. Dalam konteks ini, "*blackbox*" merujuk pada pengujian yang dilakukan dari sudut pandang eksternal, mirip dengan cara kita memandang kotak hitam yang tidak terlihat isinya. Metode ini fokus pada fungsionalitas perangkat lunak dan menguji apakah *output* yang dihasilkan sesuai dengan yang diharapkan berdasarkan *input* yang diberikan dan menguji apakah perangkat lunak berperilaku sebagaimana mestinya.



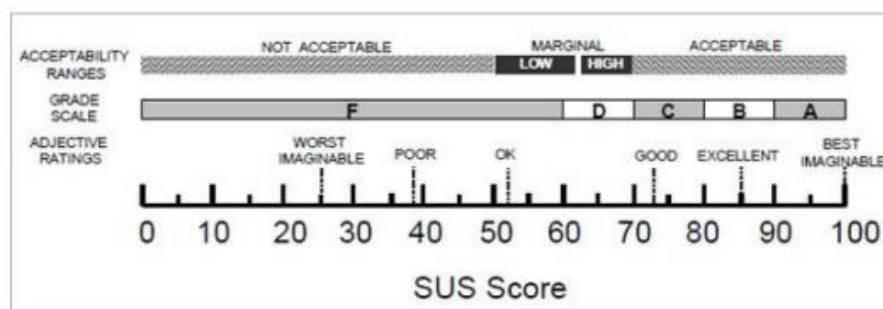
Gambar 1. Black Box Testing

Secara konseptual, *blackbox testing* bisa dibandingkan dengan pengujian sebuah mobil dengan hanya menggunakan pengendaraannya tanpa membuka kap mesin atau memahami bagaimana mesinnya bekerja. Tujuannya adalah memastikan mobil berfungsi dengan baik dari sudut pandang pengguna, tanpa memperhatikan detail teknis bagaimana mesin bekerja di dalamnya.

2.1.10 *System Usability Scale (SUS)*

System Usability Scale (SUS) umumnya digunakan menentukan tingkat *usability* atau kebergunaan dari suatu sistem. Metode SUS dapat digunakan setelah responden menggunakan sistem yang akan dievaluasi. 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 harus diikuti. Aturan-aturan tersebut meliputi: setiap jawaban dinilai dalam interval 0-4, di mana nilai 4 adalah jawaban terbaik. Untuk pertanyaan bernomor ganjil (1, 3, 5, 7, 9), nilai skala jawaban dikurangi dengan 1, sedangkan untuk pertanyaan bernomor genap (2, 4, 6, 8, 10), perhitungannya adalah 5 dikurangi skala jawaban responden [19]. Setelah itu, jumlah keseluruhan jawaban setiap responden dikalikan dengan 2,5. Selanjutnya, dihitung nilai rata-rata dari semua responden.



Gambar 2. Standar penilaian SUS

Gambar 2 menerangkan mengenai SUS yang menggunakan tiga sudut pandang dalam penilaian: *acceptability range*, *grade scale*, dan *adjective ratings*. *Acceptability* digunakan untuk menilai tingkat penerimaan pengguna terhadap suatu sistem. Berdasarkan sudut pandang ini, penilaian dibagi menjadi not *acceptable*, *marginal*, dan *acceptable*. Predikat *not acceptable* diberikan jika skor akhir SUS adalah 0-50, *marginal low* untuk skor 50-60, *marginal high* untuk skor 60-70, dan *acceptable* untuk skor 70-100 [19].

2.2 Penelitian Terdahulu

Dalam melaksanakan penelitian ini, beberapa teknik pengumpulan data telah dilakukan dengan menganalisa studi literatur terhadap penelitian-penelitian terdahulu yang sesuai dan relevan dengan topik pengembangan platform pada penelitian ini. Analisa tersebut yang menjadi acuan dan referensi sekaligus pembaruan metode pengembangan yang sesuai dengan penelitian ini. Berikut merupakan beberapa penelitian terdahulu yang relevan dengan topik penelitian ini.

Penelitian yang dilaksanakan oleh Kaligis dkk., pada tahun 2020 yang berjudul *Pengembangan Tampilan Antarmuka Aplikasi Survei Berbasis Web Dengan Metode User Centered Design*. Penelitian ini memiliki persamaan pada topik yang dibawakan yaitu perancangan aplikasi survei berbasis *web*. Metode yang digunakan dalam penelitian ini adalah metode *User centered design*. Penelitian ini melibatkan pengguna dalam proses pembangunan sistem sehingga mampu menyesuaikan dengan kebutuhan pengguna. Hasil dari penelitian ini berupa data evaluasi untuk tampilan antar muka dalam aspek efektivitas, efisiensi, dan kepuasan pengguna yang dapat dimanfaatkan untuk mengembangkan aplikasi sesuai dengan kebutuhan pengguna [20].

Penelitian yang dilaksanakan oleh Nur Fajri dkk., pada tahun 2021 yang berjudul *Perancangan Pengalaman Pengguna Aplikasi Survei Online Berbayar Antar Mahasiswa berbasis Mobile menggunakan Metode Design Thinking*. Sama seperti sebelumnya, penelitian ini memiliki topik yaitu perancangan aplikasi survei *online* berbayar dan memiliki cukup banyak persamaan dengan Quision pada cara kerja sistem. Penelitian ini menghasilkan produk berupa prototype sistem dan data-data pengalaman pengguna dari hasil benchmark UEQ dengan aspek *attractiveness*,

stimulation, dan *novelty*. Pada prototype di penelitian ini terdapat mahasiswa sebagai responden yang bertujuan untuk mendapatkan *reward* dengan mengisi survei yang ada. Adapun metode penelitian yang digunakan dalam penelitian ini adalah metode *design thinking* [21].

Penelitian yang dilaksanakan oleh Adhe Rama dkk., pada tahun 2020 dengan judul *Pengembangan Sistem Pengelolaan dan Pemantauan Proyek dengan Metode Agile Pola Scrum*. Penelitian ini menggunakan *Unified Modelling Language* (UML) untuk menjabarkan fungsionalitas sistem yang ada pada perangkat lunak tersebut. Disamping itu, penelitian ini juga menggunakan metode *Agile* sebagai metode pengembangan perangkat lunaknya. Kerangka kerja yang digunakan adalah *Scrum* yang mempunyai prinsip tanggap dalam menangani perubahan. Penelitian ini memiliki kesamaan pada metode pengembangan perangkat lunak dan kerangka kerja yang digunakan pada platform Quision, yaitu metode *Agile* dengan kerangka kerja *Scrum*. Produk yang dihasilkan dari penelitian ini sistem informasi yang ditujukan untuk pengelolaan dan pemantauan proyek [10].

Penelitian yang dilaksanakan oleh Adhistya Erna dkk., pada tahun 2020 berjudul *Website Development as a Promotional Supporting Tool on Butterfly Park*. Penelitian ini menggunakan metode pengembangan *Agile* dan kerangka kerja *Scrum* dengan memanfaatkan *Unified Modeling Language* (UML). Dalam proses pengembangan aplikasi pada penelitian tersebut, peneliti menggunakan *WordPress framework* sebagai sistem utama. Integrasi antara *components*, *module*, *template* dan *plugin* telah dilakukan di penelitian ini untuk menyesuaikan dengan kebutuhan fungsionalitas yang telah dirancang. Hasil dari penelitian ini adalah suatu sistem informasi kupu-kupu yang terafiliasi dengan Alian Butterfly Park yang responsif dan menarik. Berdasarkan hal tersebut, maka dapat disimpulkan bahwa pemanfaatan *WordPress* sebagai *framework* dalam pengembangan suatu aplikasi dapat memberikan nilai positif bagi aplikasi yang dikembangkan [13].

Beberapa penelitian yang telah dijabarkan sebelumnya telah membantu peneliti dalam proses pelaksanaan penelitian ini. Acuan metode pengembangan, implementasi, pengujian, dan *framework* perangkat lunak yang digunakan pada penelitian tersebut dapat dijadikan referensi dan disesuaikan kembali dengan

kebutuhan pengembangan platform Quision berbasis aplikasi *web* yang menggunakan metode *Agile* dengan kerangka kerja *Scrum*. Penelitian ini memanfaatkan *WordPress* sebagai *framework* yang digunakan untuk mengolah dan menampilkan berbagai data yang dibutuhkan ke halaman pengguna. Platform Quision juga menggunakan *payment gateway* untuk mengintegrasikan proses pembayaran yang aman dan cepat. SMTP atau *Simple Mail Transfer Protocol* juga digunakan dalam pengiriman *email* yang terintegrasi dengan sistem Quision, membuat interaksi antar pengguna menjadi lebih sering dan meningkatkan awareness pada aplikasi Quision. Salah satu pustaka pada *WordPress* yang biasa digunakan dalam pengembangan *web* modern, yaitu *Plugins* dan *Snippets*, juga menjadi salah satu pondasi pengembangan aplikasi ini.

2.3 Analisis Kompetitor

Pada penelitian ini dilakukan analisis terhadap aplikasi yang serupa namun tidak sama, baik dari cara kerja sistem maupun sisi kelebihan dan kelemahan. Adapun beberapa aplikasi yang akan dianalisis yaitu aplikasi Menyelami. Tujuan dari pelaksanaan analisis kompetitor ini adalah untuk menjadi acuan dan referensi dalam proses pengembangan perangkat lunak Quision pada penelitian ini. Dengan begitu diharapkan aplikasi Quision pada penelitian ini memiliki unsur keterbaruan dan memenuhi kekurangan yang ada pada beberapa aplikasi lainnya sebagai kompetitor. Tentunya keterbaruan aplikasi yang dikembangkan pada penelitian ini sesuai dengan tujuan penelitian dan sistematika bisnis yang telah dirancang.

Quision dirancang sebagai sebuah inovasi dalam bentuk platform yang menghubungkan antara peneliti dan responden, platform ini mengintegrasikan teknologi untuk menciptakan sebuah wadah yang terintegrasi mulai dari *role* pengguna, panduan, pencairan, pembayaran, *tracking* pengisian kuesioner hingga penerbitan kuesioner. Quision digagas dengan tampilan *website* yang menarik dengan fitur kompleks dan canggih yang memungkinkan penghubungan langsung antara peneliti dengan responden melalui *website* resmi. Di sisi lain, Menyelami memiliki tampilan yang cukup menarik namun belum cukup baik dalam menyediakan sarana fitur yang bisa mengakomodasi dari kedua sisi baik dari sisi responden maupun peneliti. Akomodasi seperti insentif untuk responden,

pembayaran secara langsung untuk peneliti, serta *customer service* yang siap siaga belum sepenuhnya dimiliki oleh platform Menyelami. Salah satu keunggulan utama Quision adalah pemeriksaan jawaban responden secara langsung serta pemberian *reward* untuk responden, di mana responden mendaftar di platform, mengisi data diri, melakukan pengisian kuesioner, dan mendapatkan *reward* sedangkan responden yang mengisi kuesioner pada platform Menyelami beresiko menjawab dengan rancu serta tidak diketahui data diri dan darimana asalnya.

Meskipun demikian, Quision masih memiliki kelemahan seperti belum menerima kriteria responden untuk menjawab secara tatap muka ataupun yang terlalu spesifik. Sementara itu, Menyelami telah menyediakan platform yang cukup interaktif dan dapat digunakan secara gratis oleh semua orang termasuk memberi kriteria spesifik yang diinginkan, namun memiliki keterbatasan dalam akomodasi, validasi, serta fitur-fitur lainnya. Dengan demikian, Quision menonjol sebagai pilihan yang menarik dan dirancang dengan harga yang terjangkau serta integrasi fitur yang lebih baik, meskipun masih ada beberapa aspek yang perlu diperbaiki untuk meningkatkan daya saingnya.

2.4 Tools yang Digunakan

2.4.1 WordPress

WordPress adalah salah satu platform manajemen konten (CMS) yang paling populer di dunia. Fokus dari CMS ini adalah pada kemudahan penggunaan dan fleksibilitas, WordPress dapat membuat *developer* untuk membuat dan mengelola situs *web* dengan cepat dan efisien. WordPress dibangun dari bahasa pemrograman PHP, dengan *database* MySQL dan semua aplikasi open-source [22]. WordPress menawarkan berbagai fitur yang dapat disesuaikan, seperti pemilihan tema yang luas, *plugin* yang dapat diintegrasikan, dan kemampuan untuk mengatur tata letak dan konten dengan mudah melalui antarmuka pengguna yang intuitif.

Selain kemudahan penggunaan, kekuatan WordPress juga terletak pada fleksibilitasnya. Pengguna dapat menggunakan WordPress untuk membuat berbagai jenis situs *web*, mulai dari blog pribadi hingga situs *web* bisnis besar. Dengan beragam tema dan *plugin* yang tersedia, pengguna memiliki kontrol penuh

atas penampilan dan fungsionalitas situs mereka. WordPress juga terus berkembang melalui kontribusi dari komunitas pengguna yang besar, dengan pembaruan reguler dan pemeliharaan yang memastikan keamanan dan kinerja platform tetap terjaga.

WordPress dapat diinstal dengan opsi instalasi cepat atau secara manual. Untuk menginstal WordPress secara manual, paket WordPress harus diunduh dari WordPress.org. Kemudian, pengguna harus mengekstrak folder yang diunduh di komputer lokal. Sekarang, WordPress harus diunggah ke akun *hosting*. WordPress dapat diunggah ke akun *hosting* melalui FTP, *File Manager*, atau SSH. Setelah berhasil mengunggah paket WordPress ke akun *hosting*, saatnya untuk membuat *database* MySQL dan *User*. WordPress biasanya menyimpan semua informasinya dalam *database*. Ada sebuah *wizard* yang disebut *MySQL Database Wizard* di cPanel. Dengan mengklik ikon *wizard* ini, *database* MySQL dan pengguna dapat dibuat. Penting untuk memeriksa semua kotak *privileges* agar dapat memodifikasi *database* nantinya. Terakhir, pengembang harus mengonfigurasi *file* bernama *wp-config.php* yang ada di *File Manager* di folder yang sama di mana WordPress diinstal.

2.4.2 *Plugin*

Plugin merupakan tambahan perangkat lunak atau ekstensi yang digunakan untuk memperluas atau menambah fungsionalitas suatu program atau platform. Dalam konteks pengembangan perangkat lunak, *plugin* sering digunakan untuk mengintegrasikan fitur-fitur baru, memodifikasi tampilan, atau meningkatkan kinerja program tanpa harus mengubah kode sumber asli. *Plugin* memungkinkan pengembang atau pengguna untuk menyesuaikan aplikasi sesuai kebutuhan spesifik mereka, dengan menyediakan cara yang lebih mudah dan fleksibel daripada menulis kode dari awal. *Plugin* juga dapat memberikan solusi yang efisien untuk memecahkan masalah tertentu atau menambahkan fitur yang tidak tersedia secara default dalam program tersebut. Pengembangan *plugin* telah menjadi strategi umum dalam pengembangan perangkat lunak modern, memungkinkan untuk memperluas fungsionalitas dan meningkatkan pengalaman pengembang dengan cara yang mudah diakses dan diimplementasikan. Umumnya implementasi pengembangan

plugin diterapkan untuk memperluas fungsionalitas pada pengembangan *web* dengan *framework* seperti Flask, Django, Laravel hingga WordPress,

Plugin pada WordPress adalah *additional* atau tambahan perangkat lunak yang memungkinkan pengembang untuk menambahkan fitur tambahan, mengubah tampilan, atau meningkatkan kinerja situs *web* mereka dengan mudah dan cepat. *Plugin* merupakan ekstensi yang fleksibel dan efisien, berisi kode yang digunakan untuk menjalankan *trigger event* dan digunakan untuk memformat *output* komponen atau modul saat halaman dibuat [13]. *Plugin-plugin* ini dapat melakukan berbagai tugas, mulai dari menambahkan formulir kontak hingga mengoptimalkan SEO, memungkinkan pengembang untuk memperluas fungsionalitas situs mereka tanpa harus menulis kode dari awal. Selain itu, dengan komunitas pengembang yang besar dan aktif, pengembang WordPress juga dapat dengan mudah menemukan dan menginstal *plugin* baru serta mendapatkan dukungan dan pembaruan reguler.

2.4.3 *Snippet*

Secara umum, *snippet* merujuk pada potongan kecil dari kode atau teks yang dapat diintegrasikan ke dalam program atau platform perangkat lunak untuk melakukan fungsi tertentu atau menambahkan fitur tambahan. *Snippets* dapat berupa kode pemrograman dalam berbagai bahasa seperti HTML, CSS, JavaScript, Python, dan lainnya, atau dapat berupa teks sederhana yang memberikan instruksi atau informasi spesifik. *Snippet* sering digunakan oleh pengembang perangkat lunak atau pengguna yang ingin menyesuaikan atau memperluas fungsionalitas suatu sistem tanpa harus menulis ulang kode dari awal. Representasi penggunaan *snippet* menunjukkan secara signifikan hasil pengerjaan yang lebih cepat dan efisien serta integrasi kode yang masih bisa diskalakan secara berkelanjutan [23]. *Snippets* sering digunakan untuk mengimplementasikan logika bisnis, menambahkan efek visual, atau mengimplementasikan fitur-fitur khusus dengan cepat. Implementasi tersebut bisa diterapkan untuk berbagai *framework* masa kini seperti Laravel, Django, Ruby on Rails, Express.js, React, Vue.js, Angular hingga WordPress.

Perkembangan implementasi *Snippets* dalam konteks WordPress merujuk pada potongan kode atau teks yang disisipkan ke dalam tema atau *plugin* WordPress untuk menambahkan atau mengubah fungsionalitas situs *web*. *Snippets* ini dapat berupa segala hal, mulai dari perubahan kecil dalam tata letak atau desain hingga integrasi fitur-fitur baru yang tidak tersedia secara default. WordPress *developer* umumnya menggunakan *snippets* untuk menyesuaikan situs mereka sesuai kebutuhan spesifik, tanpa harus mengubah secara besar-besaran kode sumber asli. Informasi mengenai *snippets* dapat di forum atau situs *web* tertentu yang didedikasikan untuk berbagi informasi dan pengetahuan tentang WordPress. *Snippets* pada WordPress juga bisa menjadi cara efektif untuk memperluas fungsionalitas situs *web* tanpa perlu menulis kode dari awal dan bisa digunakan untuk menambahkan fitur-fitur khusus seperti *widget* tambahan, integrasi dengan layanan pihak ketiga, atau penyesuaian logika bisnis tertentu. Dengan menggunakan *snippets*, pengguna dapat dengan cepat mengimplementasikan perubahan atau tambahan yang mereka inginkan tanpa harus memahami sepenuhnya struktur internal WordPress atau memiliki keterampilan pemrograman yang sangat maju.

2.4.4 *Payment gateway*

Payment gateway merupakan sistem yang memfasilitasi transaksi pembayaran *online* dengan menghubungkan situs *web* atau aplikasi *e-commerce* dengan lembaga keuangan, seperti bank atau penyedia layanan pembayaran. Fungsi utama dari *payment gateway* adalah untuk memproses informasi pembayaran secara aman dan efisien, mulai dari pengambilan detail kartu *credit* atau informasi pembayaran lainnya hingga otorisasi transaksi dan pemrosesan pembayaran. *Payment gateway* juga memainkan peran penting dalam keamanan transaksi *online* dengan menyediakan enkripsi data dan memastikan bahwa informasi sensitif pelanggan disimpan dan ditangani dengan aman. Penggunaan *payment gateway* sebagai alat transaksi dapat membuat usaha bisnis lebih maju karena proses pembayaran lebih mudah dan pendapatan penjualan menjadi meningkat [24]. Bisnis dapat menerima berbagai jenis pembayaran *online*, seperti kartu *credit*, transfer bank, atau pembayaran digital, dan memperluas jangkauan mereka ke pasar global

dengan menyediakan opsi pembayaran yang nyaman dan aman bagi pelanggan mereka.

Di Indonesia, terdapat berbagai *payment gateway* yang menghubungkan bisnis dengan pelanggan untuk memproses pembayaran secara *online* dengan mudah dan aman. Salah satu provider yang paling populer adalah Xendit. Xendit adalah *payment gateway* yang menyediakan solusi pembayaran digital yang lengkap, mulai dari pembayaran kartu *kredit*, transfer bank, hingga pembayaran digital seperti *e-wallet* dan *virtual account*. Setiap transaksi di *payment gateway* Xendit dapat dengan mudah direkonsiliasi dan dicatat melalui deteksi dana otomatis dengan pemberitahuan/notifikasi yang instan, integrasi yang mudah, fitur keamanan yang canggih, dan dukungan pelanggan yang responsif [25].

Payment gateway Xendit juga bisa diintegrasikan pada sistem WordPress sehingga dapat digunakan untuk memproses pembayaran secara *online* dengan lancar dan aman. Pengembang dapat dengan cepat dan efisien menghubungkan situs *web* dengan layanan pembayaran Xendit dengan menggunakan *plugin* resmi Xendit untuk WordPress. *Plugin* ini menyediakan fitur-fitur seperti pembayaran dengan kartu *kredit*, transfer bank, *e-wallet*, dan *virtual account*, serta dukungan untuk pembayaran berulang (*recurring payments*) dan pembayaran satu kali (*one-time payments*). Pengembang WordPress dapat memperluas fungsionalitas situs *web* dengan pembayaran *online* yang terintegrasi secara langsung, meningkatkan kenyamanan bagi pelanggan dan mengoptimalkan proses bisnis secara keseluruhan.

2.4.5 Simple Mail Transfer Protocol (SMTP)

Simple Mail Transfer Protocol (SMTP) adalah protokol komunikasi yang digunakan untuk mengirim *email* melalui jaringan komputer, terutama Internet. Protokol ini bertanggung jawab untuk mentransfer *email* dari pengirim ke *server email* penerima, serta mengatur komunikasi antara *server email* yang berbeda. SMTP berfungsi sebagai jembatan untuk pengiriman *email* dengan meneruskan pesan dari klien *email* atau aplikasi *email* ke *server email* penyedia layanan, yang kemudian mengirimkan pesan tersebut ke tujuan akhir. Selain mengirim *email*, SMTP juga berperan dalam otentikasi pengguna, pengaturan prioritas pengiriman,

dan penanganan pesan bounces (pesan yang gagal dikirim). SMTP dapat mendukung beberapa jenis pengiriman pesan yaitu pengiriman pesan pada satu atau lebih pengguna, pengiriman pesan yang termasuk didalamnya teks, suara, video atau gambar, pengiriman pesan ke pengguna-pengguna yang diluar jaringan atau internet [26]. SMTP bekerja bersama dengan protokol POP3 atau IMAP yang digunakan untuk menerima *email*. Dengan menggunakan SMTP, pengguna dapat mengirim *email* dengan cepat dan andal, menjadikannya inti dari sistem komunikasi *email* yang digunakan secara luas di seluruh dunia.

Integrasi SMTP memungkinkan aplikasi atau platform untuk mengirim *email* secara langsung melalui *server email* dengan menggunakan suatu protokol. Memungkinkan pengguna untuk mengirim *email* dari dalam aplikasi mereka tanpa perlu mengandalkan klien *email* eksternal. Contoh integrasi SMTP yang umum termasuk layanan seperti SendGrid, Mailgun, Brevo, dan Amazon SES (*Simple Email Service*). Brevo adalah aplikasi manajemen hubungan pelanggan (CRM) yang mengintegrasikan SMTP untuk mengirim *email* otomatis kepada pelanggan, seperti *email* pemberitahuan, konfirmasi pesanan, atau pesan promosi [27]. Brevo juga menyediakan integrasi dengan sistem WordPress dan dapat menyediakan fitur komunikasi yang kuat kepada *Users* di sistem tersebut, memperkuat hubungan dengan pelanggan, dan meningkatkan kinerja bisnis model secara keseluruhan.

2.4.6 *Application Programming Interface (API)*

Application Programming Interface (API) merupakan antarmuka yang terdiri dari gabungan perintah yang menjelaskan bagaimana agar suatu unit software dapat berkomunikasi dengan unit software lainnya. Tujuan dibangunnya API adalah untuk mempersingkat proses pengembangan dengan menyediakan serangkaian fungsi atau layanan yang dapat diakses oleh *developer* untuk digunakan dalam pengembangan perangkat lunak [28]. API dapat membuat sistem bertransaksi data, mengakses fungsionalitas tertentu, atau bahkan berinteraksi dengan sistem eksternal melalui permintaan dan respons yang telah diprogram dengan jelas.

Dari segi implementasi, API sering kali diwakili oleh serangkaian endpoint atau titik akhir yang menyediakan akses ke fungsi atau data tertentu. Penggunaan

API dapat melibatkan berbagai metode komunikasi, seperti protokol HTTP dan HTTPS, serta *format* data yang umum seperti JSON atau XML untuk pertukaran informasi. Pengembang menggunakan dokumentasi API untuk memahami cara penggunaan API, termasuk parameter yang diperlukan, *format* permintaan dan respons, serta kemungkinan respons yang dapat diterima. Selain itu, API juga sering digunakan untuk mengintegrasikan berbagai sistem dan layanan, memungkinkan aplikasi untuk beroperasi dengan lebih efisien dan terhubung dengan lingkungan teknologi yang lebih luas.

API juga memainkan peran penting dalam pembangunan ekosistem perangkat lunak secara terbuka dan terhubung. Menyediakan sistem dengan API yang terbuka dan dokumentasi yang baik dapat membuat *developer* bisa mempercepat proses adaptasi dan komunikasi tim *developer* satu dengan lainnya sehingga dapat memperluas cakupan dan potensi pengembangan aplikasi. Selain itu, dalam penggunaan API, penting untuk memperhatikan keamanan dan privasi data, serta mengimplementasikan praktik-praktik terbaik dalam desain dan pengembangan API untuk memastikan keandalan, skalabilitas, dan kinerja yang optimal. Dengan demikian, API bukan hanya merupakan alat teknis, tetapi juga merupakan elemen kunci dalam strategi untuk melakukan pengembangan perangkat lunak.

2.4.7 MySQL

MySQL adalah sistem manajemen basis data relasional (RDBMS) yang populer dan *open-source*. MySQL telah menjadi salah satu pilihan utama untuk aplikasi *web*, aplikasi bisnis, dan berbagai aplikasi lainnya yang membutuhkan penyimpanan dan pengelolaan data yang efisien. MySQL didukung oleh bahasa pemrograman SQL (*Structured Query Language*) yang bisa membuat pengguna untuk melakukan berbagai operasi, seperti pencarian, penyaringan, pengurutan, dan manipulasi data dalam *database* [29].

Secara teknis, MySQL menggunakan model *client-server*, di mana klien dapat terhubung ke *server* MySQL melalui jaringan untuk mengakses dan memanipulasi data dalam *database*. *Server* MySQL berfungsi sebagai pengelola basis data yang menerima permintaan dari klien, menjalankan perintah SQL yang

sesuai, dan mengirimkan hasilnya kembali ke klien. MySQL mendukung berbagai jenis penyimpanan data, termasuk tabel, indeks, tampilan, prosedur tersimpan, dan fungsi, yang memungkinkan pengguna untuk menyimpan dan mengelola data dengan berbagai cara sesuai dengan kebutuhan aplikasi mereka. Salah satu keunggulan utama MySQL adalah keandalannya, kinerja yang baik, dan skalabilitas yang mudah diperluas. MySQL dapat diinstal dan dijalankan di berbagai platform, termasuk Windows, Linux, macOS, platform *cloud* seperti Amazon *Web Services* (AWS) dan Google *Cloud Platform* (GCP), dan *framework* seperti Laravel, Ruby, Django, bahkan CMS seperti Drupal, Joomla, dan WordPress.

Penggunaan MySQL berperan penting dalam ekosistem WordPress. WordPress, yang dikembangkan menggunakan bahasa pemrograman PHP, memanfaatkan MySQL sebagai sistem manajemen basis data untuk menyimpan, mengelola, dan mengakses konten situs *web*, termasuk entri blog, halaman, pengguna, komentar, dan lain sebagainya [30]. Selain itu, MySQL memiliki komunitas yang besar dan aktif, yang menyediakan dukungan, sumber daya, dan pengembangan terus-menerus untuk memastikan bahwa MySQL tetap menjadi salah satu sistem basis data relasional yang paling populer dan andal di dunia.

2.4.8 Draw.io

Draw.io adalah sebuah perangkat lunak yang diperuntukkan pengguna untuk membuat berbagai jenis diagram secara *online* dengan akses mudah dan cepat melalui berbagai perangkat tanpa perlu menginstal perangkat lunak tambahan. Dari segi arsitektur, draw.io menggunakan pendekatan berbasis komponen, di mana berbagai elemen diagram seperti bentuk, panah, dan teks diimplementasikan sebagai komponen independen yang dapat dimanipulasi oleh pengguna dan memiliki fleksibilitas dalam merancang diagram yang sesuai dengan kebutuhan spesifik pengguna. Selain itu, Draw.io juga memiliki kemampuan pembuatan pemodelan, tata letak, dan kolaborasi serta memfasilitasi pembuatan prinsip-prinsip teknologi yang kompleks. Fitur utamanya mencakup penyimpanan model yang dibuat dalam berbagai *format*, pembuatan berbagai diagram basis data seperti diagram UML, serta ketersediaan template untuk desain basis data. Selain itu, platform ini mendukung kerja bersama, memungkinkan kolaborasi tim dalam

pembuatan dan penyuntingan diagram [31]. Draw.io juga menyediakan fitur untuk menampilkan hubungan antar tabel, serta pengguna dapat melakukan untuk mengekspor diagram basis data dalam berbagai *format* seperti gambar, PDF, HTML, dan XML.

2.4.9 Figma

Figma adalah sebuah platform perangkat lunak yang diperuntukkan untuk mendesain secara bersama-sama dalam suatu tim mulai dari proses desain, prototyping, hingga pengembangan produk digital. Figma dapat memberikan pengalaman pengguna yang responsif dan cepat melalui browser *web* tanpa memerlukan instalasi perangkat lunak tambahan. Tujuan utama dari pembuatan Figma adalah untuk pengguna dapat mendesain produk digital, dan dapat berkolaborasi dengan anggota tim untuk bekerja secara efisien dan efektif, terlepas dari lokasi atau perangkat yang digunakan. Figma juga menyediakan lingkungan yang terintegrasi untuk desain dan *prototyping*, memungkinkan pengguna untuk membuat, mengedit, dan berbagi desain serta prototipe dalam satu tempat. Cara kerja Figma didasarkan pada konsep *cloud computing*, di mana semua proyek dan aset desain disimpan secara aman di *cloud* dan dapat diakses dari mana saja dengan koneksi internet. Selain itu, software *editing* figma dapat digunakan untuk merancang tampilan aplikasi analytics yang modern, minimalis, dan mudah untuk digunakan [32].

2.4.10 PHP (*Hypertext Preprocessor*)

PHP atau *Hypertext Preprocessor* adalah bahasa pemrograman sisi *server* yang digunakan secara luas untuk pengembangan aplikasi *web* dinamis dan interaktif. Penggunaan PHP umumnya menyatu dengan HTML dan dijalankan di *server web* lalu digunakan untuk menghasilkan konten *web* secara dinamis yang dikirimkan ke sisi browser [33]. Bahasa ini memiliki sintaksis yang mirip dengan bahasa pemrograman C dan Perl, dan mendukung berbagai fitur pemrograman seperti variabel, *loop*, fungsi, dan objek. Salah satu keunggulan utama PHP adalah kemampuannya untuk terintegrasi dengan berbagai basis data, seperti MySQL,

PostgreSQL, dan MongoDB. PHP tetap menjadi salah satu bahasa pemrograman yang sangat relevan dan populer dalam industri teknologi karena fleksibilitas, kemudahan penggunaan, dan kekuatan dalam pengembangan *web*.

PHP memainkan peran yang sangat penting dalam sistem WordPress karena WordPress sendiri dibangun dengan menggunakan bahasa pemrograman PHP. PHP digunakan untuk mengelola dan memproses konten dinamis yang ditampilkan dalam halaman-halaman WordPress [30]. Selain itu, PHP juga digunakan untuk mengontrol logika bisnis di balik fitur-fitur WordPress, seperti manajemen pengguna, halaman, dan pengaturan situs. Karena tingkat fleksibilitas yang tinggi, pengembang WordPress dapat dengan mudah menyesuaikan situs dengan menulis kode tambahan atau membuat *plugin* untuk menambahkan fungsionalitas khusus sesuai kebutuhan mereka dengan menggunakan PHP.

III. METODOLOGI PENELITIAN

3.1 Waktu dan Tempat

3.1.1 Waktu

Penelitian ini merupakan kegiatan yang berhasil lolos ke dalam puncak tahapan Program Kreativitas Mahasiswa Kewirausahaan (PKM-K) yaitu menjadi salah satu dari bagian Pekan Ilmiah Mahasiswa Nasional (PIMNAS) ke 36. Penelitian ini membahas pembuatan platform hingga tahap platform beroperasi dan dapat dipasarkan, sesuai dengan tenggat waktu yang diberikan yaitu tiga bulan setelah pendanaan. Untuk mendapatkan hasil *output* berupa platform Quision yang siap dipasarkan, penelitian ini telah dilaksanakan selama kurang lebih 3 bulan mulai dari tanggal 15 Juni 2023 hingga 15 September 2023. Namun, secara umum pengembangan platform Quision telah berlangsung selama 11 bulan mulai dari bulan Juni 2023 hingga April 2024 yang terlihat pada tabel berikut.

Tabel 3. Waktu pelaksanaan penelitian

[illegible]

3.1.2 Tempat

Penelitian ini dilaksanakan di lingkungan Laboratorium Teknik Digital, Program Studi Teknik Infomatika, Jurusan Teknik Elektro, Fakultas Teknik, Universitas Lampung, dan bertempat di Gedong Meneng, Kec. Rajabasa, Kota Bandar Lampung, Lampung.

3.2 Alat dan Bahan Penelitian

3.2.1 Alat Penelitian

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

Tabel 4. Perangkat keras

Perangkat Keras	Fungsi
Komputer AMD Ryzen 3 2200G	Melakukan perancangan dan implementasi algoritma pada platform Quision
Laptop Intel Core i3 gen 10	Melakukan perancangan dan implementasi algoritma pada platform Quision
<i>Mouse</i>	Menggerakan kursor
<i>Keyboard</i>	Mengetik kata atau kalimat

Tabel 5. Perangkat lunak

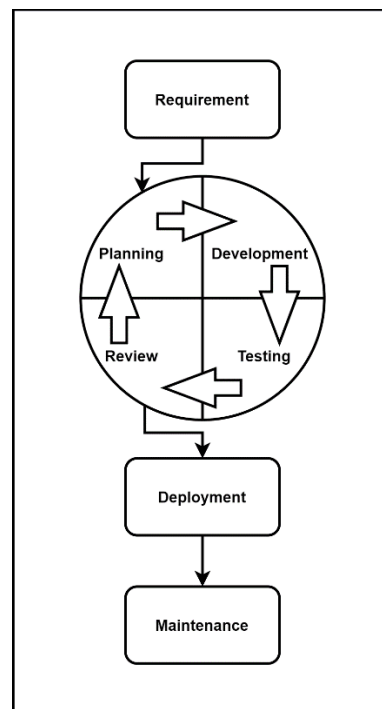
Perangkat Lunak	Keterangan
Google chrome (<i>Browser</i>)	Media implementasi <i>web</i>
WordPress	<i>Framework</i> pengembangan platform
<i>Plugin</i>	Media integrasi ekstensi fitur platform
<i>Snippet</i>	Media integrasi <i>coding</i> platform
Niagahoster Hpanel (<i>Hosting</i>)	Penyedia layanan <i>hosting</i>
Draw.io	<i>Tools</i> untuk membuat diagram
Figma	<i>Tools</i> untuk membuat desain halaman
PHP (<i>Hypertext Preprocessor</i>)	Bahasa pemrograman pengembangan <i>web</i>

3.2.2 Bahan Penelitian

Pada penelitian ini, terdapat beberapa bahan penelitian yang digunakan untuk mendukung informasi, metode dan proses pengembangan platform Quision.

1. Beberapa penelitian pengembangan fitur pada WordPress yang relevan.
2. Analisis kompetitor berupa referensi untuk pembuatan layanan dan fitur-fitur pengguna.
3. Jurnal penelitian yang digunakan sebagai acuan metode pengembangan platform.

3.3 Tahapan Penelitian



Gambar 3. Tahapan *Scrum*

Gambar 3 merupakan berbagai tahapan pengembangan metode *Agile* dengan kerangka kerja *Scrum* yang akan dilaksanakan dalam penelitian ini. Tahapan tersebut terdiri atas *requirements*, *planning*, *Development*, *testing*, *review*, *deployment*, dan *maintenance*.

3.3.1 *Requirements*

Pada tahap *requirements*, peneliti membuat daftar kebutuhan fungsional maupun non-fungsional yang terkait dengan ekosistem Quision yang akan dikembangkan. Kebutuhan tersebut disesuaikan dengan kebutuhan pengguna baik dari peneliti maupun responden yang diambil dari observasi secara berkala. Teknik observasi juga dilakukan dalam proses pengumpulan data untuk meninjau serta menentukan *framework* sistem, desain sistem, *timeline* secara *general*, dan fitur-fitur yang dibutuhkan saat proses pengembangan. *Framework* sistem yang dipilih dalam proses pengembangan platform Quision didasari dari tingkat kecepatan pembuatan, integrasi dengan berbagai ekstensi, keamanan yang selalu diperbarui, dan harus bisa memenuhi *timeline* yang telah ditentukan secara *general*.

Peneliti juga melakukan observasi dengan menganalisa terkait layanan, alur pemesanan, cara kerja, dan sistematika pada beberapa platform kompetitor. Kompetitor yang dianalisa dalam penelitian ini adalah platform menyelami. Alasan pemilihan beberapa aplikasi ini sebagai kompetitor adalah karena aplikasi ini menawarkan layanan jasa yang sama yaitu pemesanan responden untuk pengisian data peneliti. Beberapa aplikasi tersebut dilakukan observasi dari berbagai aspek, di antaranya adalah kelebihan, kekurangan, ulasan, fleksibilitas, dan analisis SWOT (*Strong, Weakness, Opportunity, Threats*). Penggunaan analisis dengan menggunakan metode SWOT dalam analisis kompetitor bertujuan untuk membantu memahami posisi keunggulan dan kelemahan kompetitor serta peluang dan ancaman.

Pada tahap ini juga dilakukan pembuatan *Unified Modeling Language* (UML) untuk memperjelas fungsional perangkat lunak yang akan dikembangkan. Pembuatan UML ini tentunya akan dilakukan setelah perencanaan dan analisa fitur telah ditentukan, sesuai, dan mencapai kesepakatan. Diagram *Use Case* akan digunakan untuk memberikan penjelasan hubungan antara aktor atau pengguna dengan sistem yang bertujuan untuk memberikan kemudahan bagi para pembaca dalam memahami bagaimana sistem ini bekerja dan berinteraksi dengan pengguna. Selain itu, pembuatan diagram *Use Case* ini adalah untuk memberikan kemudahan dalam proses pengembangan platform Quision pada penelitian ini.

3.3.2 *Planning*

Pada tahap *planning*, peneliti melakukan penyusunan terhadap serangkaian tugas (*Backlog*) dengan memilih fitur-fitur apa saja yang akan dikerjakan. Penentuan fitur dilaksanakan dengan cara membagi berbagai *User stories* yang telah dibuat sebelumnya menjadi beberapa kelompok. Di tahap ini juga dilakukan penggunaan *story point* yang bertujuan untuk menentukan timeline dan perkiraan waktu yang dibutuhkan dalam pengembangan setiap fitur pada iterasi tersebut. Pemilihan fitur-fitur yang akan dikerjakan juga mengacu pada skala prioritas, keterkaitan fungsional dan bobot pekerjaan yang telah ditentukan. Pengelompokkan fitur nantinya akan menyesuaikan pada banyaknya iterasi yang dilakukan dengan tiap iterasi memakan waktu berkisar satu hingga empat minggu pengerjaan.

3.3.3 *Development*

Pada tahap *Development*, peneliti bersama tim mulai mengerjakan fitur-fitur yang telah ditentukan sebelumnya. Pada tahap ini dilakukan perancangan algoritma fitur, pemilihan *tools* dan ekstensi, implementasi kode, dan penyederhanaan kode. Perancangan algoritma fitur diawali dengan mengatur skema logika dan parameter kode yang tepat pada fitur tersebut. Setelah algoritma telah dibuat, langkah selanjutnya adalah peneliti memilih *tools* dan ekstensi untuk melakukan implementasi logika tersebut ke dalam sebuah kode. Setelah kode telah terintegrasi dengan baik, peneliti menerapkan *clean code* dan mencoba fungsionalitas fitur yang telah dibuat. Pada tahap ini, pengerjaan tugas akan selalu mengacu pada *timeline* yang telah dibuat dengan harapan agar proses pengerjaan tidak melebihi batas waktu yang telah ditentukan.

3.3.4 *Testing*

Pada tahap *testing*, pengujian sistem dilakukan terhadap seluruh fitur yang telah dikembangkan. Hal ini bertujuan untuk memastikan setiap fitur dapat berjalan sesuai dengan fungsinya. Pengujian ini menggunakan metodologi *Black Box Testing*. 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. Selain itu, peneliti juga melakukan perancangan *scenario* yang bertujuan untuk menguji setiap fitur yang ada. Setelah *scenario* telah ditentukan, selanjutnya *test case* dibuat dan diuji dengan melakukan perintah sesuai pada skenario pengujian. Lalu pada tahap hasil pengujian, peneliti membuat catatan hasil pengujian dan melakukan *fixing bug* jika diperlukan.

3.3.5 Review

Pada tahap *review*, peneliti menyampaikan representasi *Backlog* yang telah diselesaikan melalui proses *sprint*. Proses *review* mengharuskan peneliti untuk melakukan demonstrasi terkait fitur-fitur yang dikembangkan dan dievaluasi langsung oleh tiap tim. Hal ini dilakukan agar bisa mendapatkan umpan balik terkait apa yang telah dikerjakan. Apabila terdapat revisi ataupun fitur-fitur yang belum diselesaikan di iterasi tersebut maka fitur tersebut akan dikerjakan pada siklus *sprint* selanjutnya.

3.3.6 Deployment

Setelah melalui proses pengembangan, *testing*, dan *review* secara berkala, tahap selanjutnya adalah *deployment*. Tahap *deployment* akan dilakukan ketika tim telah memastikan semua fitur telah diselesaikan dengan baik dan sesuai kebutuhan. Pada tahap ini, platform Quision yang telah dikembangkan akan dipublikasikan dengan resmi secara *online* dan *live* sehingga dapat diakses oleh *end user* atau pengguna baru Quision. Proses *deployment* dilakukan dengan mempublikasikan semua fitur yang telah diimplementasikan dan mengelola kode, ekstensi, *database*, dan *file-file* terkait ke *server hosting* dengan menggunakan hpanel Niagahoster. Sebelum melakukan *deployment*, peneliti dan tim memastikan bahwa *server hosting* telah terkonfigurasi untuk *framework* WordPress dan memenuhi semua persyaratan teknis yang diperlukan oleh platform Quision.

3.3.7 Maintenance

Pada tahap *maintenance*, peneliti berkewajiban untuk terus memantau dan memelihara platform agar tetap berjalan dengan baik dan memenuhi kebutuhan

pengguna. Hal ini meliputi pemantauan kinerja *server*, pengolahan *database*, dan penanganan *bug* atau masalah yang mungkin menampilkan setelah proses *deployment* dilakukan. Pada tahap *maintenance*, proses pemeliharaan yang dilakukan juga mencakup pembaruan fitur dan penyesuaian fungsionalitas yang didapat berdasarkan umpan balik pengguna. Proses pengerjaan fitur baru serta pembaruan teknologi juga tetap melibatkan proses *planning*, *development*, *testing*, dan *review* yang mengacu pada skema *Scrum* agar pengembangan platform tetap efektif dan efisien. Pada tahap ini, peneliti dan tim akan melakukan observasi serta menerima umpan balik dari pengguna secara berkala untuk memahami kebutuhan mereka dan memastikan bahwa platform Quision terus berkembang dan memenuhi harapan pengguna.

V. PENUTUP

5.1 Kesimpulan

Berdasarkan penelitian yang telah dilakukan, maka dapat disimpulkan bahwa:

1. Platform Quision telah sukses dikembangkan melalui metode pengembangan *Agile* dengan kerangka kerja *Scrum* dalam kurung waktu 84 hari dengan 6 iterasi. Pelaksanaan waktu yang diperlukan untuk menyelesaikan platform ini sesuai dengan perkiraan yang telah direncanakan sebelumnya saat tahap perencanaan.
2. Implementasi teknologi *Back End* pada Platform Quision telah mampu membantu memfasilitasi proses pemesanan responden untuk peneliti, pengisian kuesioner oleh responden, dan menyediakan fitur serta layanan transaksi bagi peneliti sehingga platform ini memiliki nilai ekonomi.
3. Berdasarkan hasil pengujian menggunakan metode *System Usability Scale* (SUS) didapatkan skor akhir dengan nilai 77,3 melalui penilaian oleh 50 responden. Hal ini menandakan bahwa platform yang telah dikembangkan sudah memenuhi nilai kebergunaan atau *usability* yang dapat diterima (*acceptable*) oleh pengguna.
4. Berdasarkan hasil pengujian menggunakan metode *Black Box testing* menunjukkan bahwa platform yang telah dikembangkan memiliki kinerja yang baik dan memenuhi kebutuhan pengguna. Adapun fitur yang telah diuji berjumlah 69 fitur dengan total keseluruhan pengujian dapat diterima dan sesuai harapan.

5.2 Saran

Adapun saran untuk penelitian selanjutnya adalah sebagai berikut.

1. Pengembangan fitur dan peningkatan layanan, salah satunya dapat membuat layanan baru yang bisa memfasilitasi peneliti untuk dapat melakukan *interview* tatap muka dengan responden.

2. Perlu adanya peningkatan dari segi automasi, salah satunya dapat memanfaatkan teknologi *Artificial Intelligence* (AI) sehingga sistem dapat mengoreksi secara otomatis apabila ada kesalahan pengisian kuesioner oleh responden.

DAFTAR PUSTAKA

- [1] Sitepu, E. M. Riska, J. A. Nainggolan and R. A. Lumbansiantar, "Urgensi Bagi Pendidikan di Negera Indonesia yang sedang Berkembang," *Jurnal Edukasi Nonformal*, vol. 4, no. 1, pp. 100-108, 2023.
- [2] Sukendra and Atmajaya, *Instrumen Penelitian*, Lumajang: Mahameru Press, 2020.
- [3] Pozzo, Borgobello and Pierella, "Using Questionnaires in Research on Universities: Analysis of Experiences from a Situated Perspective," *Universitat de Barcelona Institut de Desenvolupament Professional*, vol. 12, no. 2, pp. 1-16, 2019.
- [4] F. Deri, "Teknik Pengambilan Sampel Umum dalam Metodologi Penelitian: Literature Review," *Jurnal Ilmiah Pendidikan Holistik (JIPH)*, vol. 1, no. 2, pp. 85-114, 2022.
- [5] Prayoga, H. Zidan and I. Permadi, "Development of Final Study Management System Backend Integrated SIA in Faculty of Engineering University of General Soedirman," *Jurnal Teknik Informatika (Jutif)*, vol. 3, no. 2, pp. 447-456, 2022.
- [6] Qadafi and Wahyudi, "Sistem Informasi Inventory Gudang Dalam Ketersediaan Stok Barang Menggunakan Metode Buffer Stok," *Jurnal Informatika Dan Rekayasa Perangkat Lunak*, vol. 1, no. 2, p. 174-182, 2020.
- [7] Waja, Gopalkrishna, Nanavati, J. Shah and Pankti, "Agile Software Development," *International Journal of Engineering Applied Sciences and Technology*, vol. 5, no. 2, pp. 73-78, 2021.
- [8] Melyani, Rosita and Aji, "Pengembangan Sistem Informasi Penggajian Berbasis Web Menggunakan Framework Laravel dengan Metode Agile Software Development," *Jurnal Sistem Informasi Akuntansi (JASIKA)*, vol. 3, no. 1, pp. 31-36, 2023.
- [9] P. Aggarwal and R. M. Chandani, "Agile Methodology Influence on SDLC (Software Development Life Cycle)," 2020.
- [10] Febrianto, A. Rama, A. Wulansari and Latipah, "Pengembangan Sistem Pengelolaan dan Pemantauan Proyek dengan Metode Agile Pola Scrum," *Jurnal Teknik Informatika Dan Sistem Informasi*, vol. 6, no. 2, 2020.
- [11] Andipradana, Aryanata and K. D. Hartomo, "Rancang Bangun Aplikasi Penjualan Online Berbasis Web Menggunakan Metode Scrum," *Jurnal Algoritma*, vol. 18, no. 1, pp. 161-172, 2021.
- [12] Pratama and P. A. Eka, "Pengujian performansi lima back-end javascript framework menggunakan metode GET dan POST," *Jurnal Resti (Rekayasa Sistem dan Teknologi Informasi)*, vol. 4, no. 6, pp. 1216-1225, 2020.

- [13] Permanasari, A. Erna and dkk., "Website Development as a Promotional Supporting Tool on Butterfly Park," in *Proceedings of the International Conference on Innovation in Science and Technology (ICIST 2020)*, Yogyakarta, Atlantis Press, 2021, pp. 385-390.
- [14] Malius and Dani, "Sistem Informasi Sekolah Berbasis Web Pada Sekolah Dasar Negeri (SDN) 109 Seriti," *Indonesian Journal Of Education and Humanity*, vol. 1, no. 3, pp. 156-168, 2021.
- [15] S. Suriya, *UML Diagramming: A Case Study Approach*, Auerbach: Auerbach Publications, 2022.
- [16] Tuasamu, Lewaru and Idris, " Analisis Sistem Informasi Akuntansi Siklus Pendapatan Menggunakan DFD Dan Flowchart Pada Bisnis Porobico," *Jurnal Bisnis dan Manajemen (JURBISMAN)*, vol. 1, no. 2, pp. 495-510, 2023.
- [17] Kurniansyah and Sinurat, "Sistem Pendukung Keputusan Pemilihan Server Hosting dan Domain Terbaik Untuk WEB Server Menerapkan Metode VIKOR," *Jurnal Sistem Komputer dan Informatika (JSON)*, vol. 2, no. 1, pp. 14-24, 2020.
- [18] Fahrezi and Ahmad, "Pengujian Black Box Testing pada Aplikasi Inventori Barang Berbasis Web di PT. AINO Indonesia," *LOGIC: Jurnal Ilmu Komputer dan Pendidikan*, vol. 1, no. 1, pp. 1-5, 2022.
- [19] Gronier and Baudet, "Psychometric Evaluation of the F-SUS: Creation and Validation of the French Version of the System Usability Scale," *International Journal of Human-Computer Interaction*, vol. 37, no. 16, pp. 1571-1582, 2021.
- [20] Kaligis, D. Larson and R. R. Fatri, "Pengembangan Tampilan Antarmuka Aplikasi Survei Berbasis Web Dengan Metode User Centered Design," *JUST IT: Jurnal Sistem Informasi, Teknologi Informasi dan Komputer*, vol. 10, no. 2, pp. 106-114, 2020.
- [21] Maulidya, N. F. Hayyuni, H. Tolle and R. I. Rokhmawati, "Perancangan Pengalaman Pengguna Aplikasi Survei Online Berbayar Antar Mahasiswa berbasis Mobile menggunakan Metode Design Thinking," *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, vol. 5, no. 8, pp. 3356-3366, 2021.
- [22] Nasution, Nursanita and T. Raharjo, "Development of WordPress and Learn Press-Based Learning Management System (E-Learning)," *Akademika: Jurnal Teknologi Pendidikan*, vol. 12, no. 1, pp. 183-195, 2023.
- [23] Alon, Uri, M. Zilberstein, O. Levy and E. Yahav, "Learning Distributed Representations of Code," *Proceedings of the ACM on Programming Languages*, vol. 3, no. POPL, pp. 1-29, 2019.
- [24] Lestari, D. Anggun, E. D. Purnamasari and B. Setiawan, "Pengaruh payment gateway terhadap kinerja keuangan UMKM," *Jurnal Bisnis, Manajemen, Dan Ekonomi*, vol. 1, no. 1, pp. 1-10, 2020.

- [25] Rozaq, M. N. Jabar and U. Y. Oktiawati, "Implementasi Sistem Otomasi Order Mentoring Pada Marketplace Mentoring Platform Nusademy," *Journal of Applied Computer Science and Technology*, vol. 3, no. 2, pp. 199-207, 2022.
- [26] Thoriqy and A. Bilawal, "Monitoring Jaringan Dengan Spicework di CV. Mulya," *PROSISKO: Jurnal Pengembangan Riset dan Observasi Sistem Komputer*, vol. 9, no. 2, pp. 70-82, 2022.
- [27] Ilić and Ostojić, "Communication Management Using Digital Software to Increase Sales," *Ekonomika*, vol. 69, no. 3, pp. 43-55, 2023.
- [28] Barus, A. Christy, J. Harungguan and E. Manulu, "Pengujian API Website untuk Perbaikan Performansi Aplikasi Ditenun," *Journal of Applied Technology and Informatics Indonesia*, vol. 1, no. 2, pp. 14-21, 2021.
- [29] R. Ernayani, Anwar, Irmawati, Mala, Savitri, Mustakim and E. Rahajeng, *Dasar dan Teori Sistem Informasi Manajemen*, Batam: Cendikia Mulia Mandiri, 2023.
- [30] Iqbal, Muhammad, M. Noman, S. R. Talpu, A. Manzoor and M. M. Abid, "An Empirical Study of Popular Content Management System-WordPress vs Drupal vs Joomla," *An Empirical Study of Popular Content Management System-WordPress vs Drupal vs Joomla*, vol. 1, no. 1, pp. 1214-1219, 2020.
- [31] Vakaliuk, T. A., Korotun, O. V., Semerikov and S. O., "The Selection of Cloud Services for ER-Diagrams Construction in IT Specialists Databases Teaching," *CTE Workshop Proceedings*, vol. 8, no. 1, pp. 384-397, 2021.
- [32] I. Purnama, R. Fanny, S. Andy, A. Dicky and B. Mhd, "Rancangan UI/UX Aplikasi Analytics pada Toko Online Wao Sneakers Menggunakan Figma Berbasis Mobile," *Factory Jurnal Industri, Manajemen dan Rekayasa Sistem Industri*, vol. 1, no. 3, pp. 93-101, 2023.
- [33] Nurjaman and Yasin, "Konsep Desain Aplikasi Sistem Manajemen Kepegawaian Berbasis Web pada PT. Bintang Komunikasi Utama," *JISICOM (Journal of Information System, Informatics and Computing)*, vol. 4, no. 2, pp. 143-174, 2021.
- [34] Indrawati, Belluano, Harlinda, Tuasamu and Lantara, "Analisis Tingkat Kepuasan Pengguna Sistem Informasi Perpustakaan Menggunakan Pieces Framework," *ILKOM Jurnal Ilmiah*, vol. 11, no. 2, pp. 118-128, 2019.