

**PENGEMBANGAN *KNOWLEDGE BASE* BERBASIS *WEBSITE* UNTUK
PENGUNAAN *USER MANUAL* PADA BPJS KETENAGAKERJAAN
MENGUNAKAN METODE *AGILE***

(Skripsi)

Oleh:

**ASIMA BEATRICIA ZIVANKA RAJAGUKGUK
2115061116**



**FAKULTAS TEKNIK
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2025**

**PENGEMBANGAN *KNOWLEDGE BASE* BERBASIS *WEBSITE* UNTUK
PENGUNAAN *USER MANUAL* PADA BPJS KETENAGAKERJAAN
MENGUNAKAN METODE *AGILE***

Oleh

ASIMA BEATRICIA ZIVANKA RAJAGUKGUK

Skripsi

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

Pada

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



**FAKULTAS TEKNIK
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2025**

ABSTRAK

PENGEMBANGAN *KNOWLEDGE BASE* BERBASIS *WEBSITE* UNTUK PENGUNAAN *USER MANUAL* PADA BPJS KETENAGAKERJAAN MENGUNAKAN METODE *AGILE*

Oleh

ASIMA BEATRICIA ZIVANKA RAJAGUKGUK

Pengembangan sistem dokumentasi berbasis digital menjadi sangat penting dalam penyampaian informasi di lingkungan organisasi. BPJS Ketenagakerjaan sebelumnya menggunakan dokumen *user manual* dalam format *file* dengan ekstensi, seperti .pdf atau .docx yang menimbulkan kendala dalam proses pencarian, pembaruan, dan distribusi informasi. Penelitian ini bertujuan untuk mengembangkan *knowledge base* berbasis *website* sebagai sarana penyimpanan dan penyajian informasi panduan aplikasi, dengan tambahan fitur *chatbot* yang dibuat menggunakan Dialogflow Essentials guna memudahkan interaksi pengguna. Pengembangan dilakukan menggunakan metode *Agile* dengan tahapan iteratif, mulai dari analisis kebutuhan, desain sistem menggunakan UML, implementasi menggunakan *framework* CodeIgniter 4 dan Bootstrap 4, integrasi, pengujian melalui *User Acceptance Testing* (UAT), hingga *deployment*. Hasil pengujian, yaitu 82%, menunjukkan bahwa sistem dapat membantu pengguna dalam mengakses informasi secara lebih cepat, mudah, dan fleksibel. Sistem juga memberikan kemudahan bagi admin dalam mengelola *user manual* tanpa perlu menyusun ulang dokumen secara manual.

Kata kunci: *Knowledge base, user manual, chatbot, Agile, BPJS Ketenagakerjaan.*

ABSTRACT

DEVELOPMENT OF A WEBSITE-BASED KNOWLEDGE BASE FOR USER MANUAL UTILIZATION AT BPJS KETENAGAKERJAAN USING THE AGILE METHOD

By

ASIMA BEATRICIA ZIVANKA RAJAGUKGUK

The development of digital-based documentation systems has become increasingly important for effective information delivery within organizational environments. Previously, BPJS Ketenagakerjaan used user manuals in file formats such as .pdf or .docx, which posed challenges in terms of searching, updating, and distributing information. This study aims to develop a website-based knowledge base as a platform for storing and presenting application usage guides, enhanced with a chatbot feature developed using Dialogflow Essentials to facilitate user interaction. The development was carried out using the Agile method with iterative stages, including requirements analysis, system design using UML, implementation using the CodeIgniter 4 and Bootstrap 4 frameworks, integration, testing through User Acceptance Testing (UAT), and deployment. The test results, namely 82%, show that the system helps users access information more quickly, easily, and flexibly. The system also simplifies the work of administrators in managing user manuals without the need to manually recompile documents.

Keywords: Knowledge base, user manual, chatbot, Agile, BPJS Ketenagakerjaan.

Judul Skripsi

: PENGEMBANGAN *KNOWLEDGE BASE*
BERBASIS *WEBSITE* UNTUK PENGGUNAAN
USER MANUAL PADA BPJS
KETENAGAKERJAAN MENGGUNAKAN
METODE *AGILE*.

Nama Mahasiswa

: Asima Beatricia Zivanka Rajagukguk

Nomor Pokok Mahasiswa

: 2115061116

Program Studi

: Teknik Informatika

Fakultas

: Teknik



1. Komisi Pembimbing

Pembimbing Utama

Pembimbing Pendamping


Ir. Trisya Septiana, S.T., M.T., IPM.
NIP. 199009212019032025


Deny Budiyanto, S.Kom., M.T.
NIP. 199112082019031011

2. Mengetahui

Ketua Jurusan
Teknik Elektro

Ketua Program Studi
Teknik Informatika

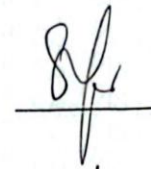

Herlinawati, S.T., M.T.
NIP. 197103141999032001


Yessi Mulyani, S.T., M.T.
NIP. 19731226200012200

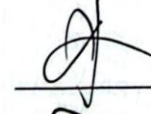
MENGESAHKAN

1. Tim Penguji

Ketua : Ir. Trisya Septiana, S.T., M.T., IPM



Sekretaris : Deny Budiyanto, S.Kom., M.T.



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



2. Dekan Fakultas Teknik



Dr. Eng. Ir. Helmy Fitriawan, S.T., M.Sc. }
NIP. 197509282001121002

Tanggal Lulus Ujian Skripsi : 29 Juli 2025

SURAT PERNYATAAN

Dengan ini saya menyatakan bahwa skripsi berjudul “ Pengembangan *Knowledge Base* Berbasis *Website* untuk Penggunaan *User Manual* pada BPJS Ketenagakerjaan Menggunakan Metode *Agile* ” sepenuhnya merupakan hasil karya saya sendiri. Apabila kemudian hari terbukti pernyataan ini tidak benar, saya siap menerima sanksi sesuai dengan ketentuan yang berlaku.

Bandar Lampung, 29 Juli 2025



Asima Beatricia Zivanka Rajagukguk

RIWAYAT HIDUP



Penulis lahir di Kotabumi pada tanggal 18 April 2003 dari pasangan Bapak Sondang Rajagukguk, S.H. dan Ibu Enny Hasan Basri, S.ST. Penulis telah menyelesaikan Pendidikan formal di SD Lentera Harapan Tulang Bawang pada tahun 2015, SMP Lentera Harapan Tulang Bawang pada tahun 2018, dan SMAN 5 Bandar Lampung pada tahun 2021. Pada tahun 2021, penulis terdaftar sebagai mahasiswi Program Studi Teknik Informatika Universitas Lampung melalui jalur SBMPTN. Selama menjadi mahasiswi, penulis telah mengikuti berbagai kegiatan dan organisasi, diantaranya :

1. Mengikuti program kampus merdeka, yaitu Studi Independen MSIB pada tahun 2023 dan menjadi TRIBE mewakili mitra sebagai narahubung antara pihak MSIB dengan setiap mahasiswa yang menjalani program MSIB pada mitra, yaitu RevoU.
2. Mengikuti program kampus merdeka, yaitu Magang MSIB pada tahun 2024 di BPJS Ketenagakerjaan sebagai *Junior Programmer*.
3. Turut serta dalam kegiatan organisasi kampus maupun luar kampus.

MOTTO

“WALK BY FAITH”

“I can do all things through Christ who strengthens me”

Philipians 4 : 13

*“Life can be heavy, especially if you try carry it all at once. Part of growing up
and moving into new chapters of your life is about catch and realese”*

-Taylor Swift

PERSEMBAHAN

Segala puji syukur bagi Tuhan Yesus Kristus, Tuhan Yang Maha Esa, karena atas hikmat dan karunia-Nya, saya selaku penulis dapat menyelesaikan skripsi ini tepat pada waktunya.

Kupersembahkan Skripsi ini Kepada:

Kedua Orang tua Tercinta

“Yang telah berjuang memenuhi kebutuhan penulis dan percaya bahwa penulis dapat menempuh pendidikan dan menyelesaikannya dengan baik”.

SANWACANA

Puji syukur ke hadirat Tuhan Yesus Kristus yang telah memberikan rahmat dan kasih-Nya kepada penulis sehingga penulis mampu menyelesaikan penelitian ini dengan baik. Skripsi berjudul “Pengembangan *Knowledge Base* Berbasis *Website* untuk Penggunaan *User Manual* pada BPJS Ketenagakerjaan Menggunakan Metode *Agile*” ini disusun sebagai salah satu syarat untuk memperoleh gelar sarjana teknik pada Program Studi Teknik Informatika, Universitas Lampung. Proses penelitian ini melibatkan banyak pihak yang membantu penulis sehingga dapat terselesaikan dengan baik. Oleh karena itu, penulis mengucapkan terima kasih yang sebesar-besarnya kepada :

1. Seluruh keluarga tercinta yang selalu ada dan memberikan dukungan sehingga penulis mampu menyelesaikan penelitian 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 yang telah membantu menyelesaikan kendala dalam proses pengerjaan penelitian.
4. Ibu Yessi Mulyani, S.T., M.T. selaku Ketua Program Studi Teknik Informatika Universitas Lampung yang telah membantu proses kelancaran pengerjaan penelitian;
5. Ibu Ir. Trisya Septiana, S.T., M.T., IPM. selaku pembimbing utama dan dosen pembimbing akademik yang telah bersedia membimbing penulis selama melaksanakan penelitian ini dan juga melaksanakan proses perkuliahan.
6. Bapak Deny Budiyanto, S.Kom., M.T. selaku pembimbing pendamping yang telah banyak memberikan masukan dalam proses penelitian ini.

7. Bapak Wahyu Eko Sulistiono, S.T., M.Sc. selaku penguji yang telah memberikan kritik dan saran yang membangun terkait pengerjaan penelitian ini.
8. Bapak Harapan Buala Daeli dan Ibu Beatrix Metaria Tambunan selaku mentor, serta karyawan BPJS Ketenagakerjaan yang telah membantu penulis dalam melaksanakan penelitian.
9. Agustin, Safira, Ghefira, Azizah, Anindya, dan Echa selaku teman seperjuangan pada masa kuliah yang selalu memberikan dukungan kepada penulis.
10. Seluruh mahasiswa Jurusan Teknik Elektro yang telah terlibat dalam proses penelitian ini yang tidak dapat penulis sebutkan satu persatu namanya.

Penulis menyadari bahwa penelitian ini masih banyak kekurangannya, baik dari segi penulisan maupun sistem yang dikembangkan. Oleh karena itu, penulis memohon maaf dan mengharapkan kritik serta saran yang membangun terkait penelitian yang telah dilaksanakan.

Bandar Lampung, 29 Juli 2025

Penulis,

Asima Beatricia Zivanka Rajagukguk

DAFTAR ISI

	Halaman
DAFTAR TABEL	xviii
DAFTAR GAMBAR	xix
BAB I. PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Tujuan	2
1.3 Rumusan Masalah	2
1.4 Manfaat Penelitian	2
1.5 Batasan Masalah.....	3
1.6 Sistematika Penulisan.....	3
BAB II. TINJAUAN PUSTAKA	5
2.1 <i>Knowledge Base</i>	5
2.2 <i>Metode Agile</i>	6
2.3 UML (<i>Unified Modelling Language</i>).....	7
2.4 <i>Chatbot</i>	11
2.5 Skala Likert	10
2.6 UAT (<i>User Acceptance Testing</i>).....	10
2.7 Aplikasi yang Digunakan.....	11
2.8 Sistem Informasi	19
2.9 Penelitian Terdahulu	19

2.9.1	Pembuatan <i>User Manual</i> Berbasis Java sebagai Tutorial Aplikasi Sistem Informasi pada Kader Posyandu Rajawali Singosari.....	19
2.9.2	<i>Platform Knowledge-Based Website</i> untuk Meningkatkan Visibilitas UMKM di Sektor Perikanan.....	20
2.9.3	Perancangan Sistem Rekomendasi Perbelanjaan Item Kursi Kayu Pada Toko <i>Online</i> Berbasis <i>Website</i> dengan Metode <i>Knowledge Based Recommendation</i>	20
2.9.4	Rancang Bangun Aplikasi <i>Knowledge Management</i> pada Pelayanan Jasa Berbasis <i>Web</i> dengan Metode <i>Agile Development</i> (Studi Kasus : PT. Cakrawala Indonesia Sejahtera).....	21
2.9.5	<i>Knowledge Based Recommendation Modeling For Clothing Product Selection Recommendation System</i>	21
2.9.6	<i>The Effect of Electronic Human Resources Management on Organizational Health of Telecommunication companies in Jordan</i>	22
2.9.7	<i>A Survey on Complex Knowledge Base Question Answering : Methods, Challenges, and Solutions</i>	23
2.9.8	<i>Agile Software Development : Methodologies and Trends</i> ...	23
2.9.9	<i>My Chatbot Companion – a Study of Human-Chatbot Relationships</i>	24
2.9.10	<i>Harmonising Knowledge for Safer Materials Via the “Nanocommons” Knowledge Base</i>	24
2.9.11	<i>Extracting a Knowledge Base of Mechanism from COVID-19 Papers</i>	25
2.9.12	<i>Global Knowledge Base for Municipal Solid Waste Management : Framework Development and Application in Waste Generation Prediction</i>	26

2.9.13	<i>Prompting as Probing : Using Language Models for Knowledge Base Construction</i>	26
2.9.14	<i>DeepKE : A Deep Learning Based Knowledge Extraction Toolkit for Knowledge Base Population</i>	27
2.9.15	<i>Introducing a Method for Intervals Correction on Multiple Likert Scales : A Case Study on an Urban Soundscape Data Collection Instrument</i>	27
BAB III. METODE PENELITIAN		29
3.1	Waktu dan Tempat Penelitian	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	32
3.3.1	<i>Requirements</i>	32
3.3.2	<i>Design</i>	37
3.3.3	<i>Implementation</i>	55
3.3.4	<i>Integration</i>	55
3.3.5	<i>Testing</i>	55
3.3.6	<i>Deployment</i>	55
3.3.7	<i>Iteration</i>	55
BAB IV. HASIL DAN PEMBAHASAN		56
4.1	<i>Design – Iterasi 1</i>	56
4.1.1	Arsitektur Informasi	62
4.1.2	<i>Use Case Diagram</i>	65
4.1.3	<i>Activity Diagram</i>	67

4.2	<i>Implementation</i>	74
4.2.1	Manajemen <i>Frontend</i>	74
4.2.2	Implementasi <i>Frontend</i>	75
4.2.3	Manajemen <i>Backend</i>	104
4.2.4	Implementasi <i>Backend</i>	105
4.3	<i>Integration</i>	134
4.3.1	Integrasi Sistem <i>Login dan Role Management</i>	134
4.3.2	Integrasi <i>Database User Manual</i>	134
4.3.3	Integrasi Fitur <i>Search</i>	135
4.3.4	Integrasi Fitur <i>Filter</i>	135
4.3.5	Integrasi <i>CRUD (Create, Read, Update, Delete)</i> dengan <i>Database</i>	135
4.3.6	Integrasi <i>History</i>	135
4.3.7	Integrasi <i>Chatbot</i> dengan <i>Dialogflow Essentials</i> menggunakan <i>API</i>	136
4.4	<i>Testing</i>	136
4.4.1	<i>User Acceptance Testing (UAT)</i>	136
4.4.2	Hasil Pengujian menggunakan Metode <i>User Acceptance Testing</i>	138
4.5	<i>Iteration</i>	139
4.5.1	Iterasi 2 (<i>Chatbot</i>)	140
4.5.2	Iterasi 3 (<i>Halaman User Manual</i>).....	144
4.6	<i>Deployment</i>	146
BAB V. KESIMPULAN DAN SARAN		150
5.1	Kesimpulan	150
5.2	Saran.....	151
DAFTAR PUSTAKA		152

LAMPIRAN	156
A. Pertemuan <i>Online</i> melalui Zoom bersama <i>Stakeholders</i>	156
B. Link Github Aplikasi <i>Knowledge Base</i>	157
C. Desain Sebelumnya.....	157

DAFTAR TABEL

	Halaman
Tabel 1. <i>Use Case Diagram</i> [7]	7
Tabel 2. <i>Activity Diagram</i> [7]	9
Tabel 3. Waktu Penelitian	29
Tabel 4. Perangkat Keras	30
Tabel 5. Perangkat Lunak	30
Tabel 6. Daftar Pertanyaan dan Jawaban Wawancara	33
Tabel 7. Kebutuhan Fungsional	35
Tabel 8. Kebutuhan Non-Fungsional	36
Tabel 9. Aktor dalam <i>Knowledge Base</i> berbasis <i>Website</i> dan Fungsinya.....	39
Tabel 10. Aktor dalam <i>Knowledge Base</i> berbasis <i>Website</i> dan Fungsinya.....	66
Tabel 11. Pertanyaan Kuesioner	136
Tabel 12. Bobot Jawaban	138
Tabel 13. Pengujian menggunakan <i>User Acceptance Testing</i>	138

DAFTAR GAMBAR

	Halaman
Gambar 1. Tahapan Metode <i>Agile</i> [7].....	6
Gambar 2. Alur Dasar Pencocokan Intent untuk Memberikan Respons ke Pengguna Akhir.....	13
Gambar 3 Bagan Proyek	19
Gambar 4. Tahapan Penelitian	32
Gambar 5. Arsitektur Sistem.....	34
Gambar 6. Use Case Diagram.....	38
Gambar 7. <i>Login</i>	40
Gambar 8. Mencari Informasi	42
Gambar 9. Mem- <i>filter User Manual</i>	43
Gambar 10. Membaca <i>User Manual</i>	44
Gambar 11. Mengelola <i>User Manual</i> : Menambahkan <i>User Manual</i>	45
Gambar 12. Mengelola <i>User Manual</i> : Meng- <i>edit User Manual</i>	46
Gambar 13. Mengelola <i>User Manual</i> : Menghapus <i>User Manual</i>	47
Gambar 14. Mengelola <i>User Manual</i> : Melihat <i>History</i>	48
Gambar 15. <i>Chatbot</i>	49
Gambar 16. <i>Logout</i>	50
Gambar 17. Desain Tampilan <i>Chatbot</i>	51
Gambar 18. Rancangan <i>Database</i>	52
Gambar 19. Database	54
Gambar A 1. Sebelum Re-desain	56
Gambar A 2. Sesudah Re-desain.....	56
Gambar B 1. Sebelum Re-desain.....	57

Gambar B 2. Sesudah Re-desain.....	57
Gambar C 1. Sebelum Re-desain.....	57
Gambar C 2. Sesudah Re-desain.....	57
Gambar C 3. Sebelum Re-desain	58
Gambar C 4. Sesudah Re-desain.....	58
Gambar D 1. Sebelum Re-desain.....	58
Gambar D 2. Sesudah Re-desain.....	58
Gambar D 3. Sebelum Re-desain.....	59
Gambar D 4. Sesudah Re-desain.....	59
Gambar E 1. Sebelum Re-desain.....	59
Gambar E 2. Sesudah Re-desain	59
Gambar F 1. Sebelum Re-desain.....	60
Gambar F 2. Sesudah Re-desain	60
Gambar G 1. Sebelum Re-desain.....	60
Gambar G 2. Sesudah Re-desain.....	60
Gambar H 1. Sebelum re-desain	61
Gambar H 2. Sesudah Re-desain.....	61
Gambar 20. Arsitektur Informasi Pengguna.....	62
Gambar 21. Arsitektur Informasi Admin	63
Gambar 22. <i>Use Case Diagram</i>	65
Gambar 23. Melihat Deskripsi Aplikasi.....	67
Gambar 24. Mengelola <i>User Manual</i> : Meng-edit Deskripsi Aplikasi.....	68
Gambar 25. Membaca <i>What's New</i>	69
Gambar 26. Mengelola <i>What's New</i> : Menambahkan <i>What's New</i>	70
Gambar 27. Mengelola <i>What's New</i> : Meng-edit <i>What's New</i>	71
Gambar 28. Mengelola <i>What's New</i> : Menghapus <i>What's New</i>	72
Gambar 29. Mengelola <i>What's New</i> : Melihat <i>History What's New</i>	73
Gambar 30. Implementasi <i>Landing Page</i>	75
Gambar 31. Implementasi <i>Login Page</i>	76
Gambar 32. Implementasi Halaman <i>Register</i>	77
Gambar 33. Tampilan <i>Email</i> Aktivasi Akun.....	77
Gambar 34. Implementasi Halaman Lupa <i>Password</i>	78

Gambar 35. Tampilan <i>Email Reset Password</i>	78
Gambar 36. Implementasi <i>Reset Password</i>	79
Gambar 37. Implementasi Halaman Utama Pengguna	80
Gambar 38. Implementasi <i>Search Page</i>	81
Gambar 39. Implementasi <i>Filter User Manual</i>	81
Gambar 40. Implementasi <i>Chatbot</i>	82
Gambar 41. Implementasi Halaman <i>User Manual</i> Pengguna.....	83
Gambar 42. Implementasi Tampilan <i>Pop-up</i> Deskripsi Aplikasi	83
Gambar 43. Implementasi Halaman Utama Admin.....	84
Gambar 44. Implementasi Halaman Buat <i>User Manual</i>	85
Gambar 45. Implementasi Tampilan <i>Pop-up</i> Berhasil Menambah <i>User Manual</i> Baru.....	86
Gambar 46. Implementasi Halaman <i>History</i>	86
Gambar 47. Implementasi Tampilan <i>Pop-up Delete History User Manual</i>	87
Gambar 48. Implementasi Tampilan <i>Pop-up</i> Berhasil <i>Delete History User Manual</i>	87
Gambar 49. Implementasi Halaman <i>User Manual</i> Admin	88
Gambar 50. Implementasi Tampilan <i>Pop-up</i> Deskripsi Aplikasi	89
Gambar 51. Implementasi Tampilan <i>Pop-up Edit</i> Deskripsi Aplikasi	89
Gambar 52. Implementasi Tampilan <i>Pop-up</i> Berhasil <i>Edit</i> Deskripsi.....	90
Gambar 53. Implementasi Halaman <i>Edit User Manual</i>	91
Gambar 54. Implementasi Tampilan <i>Pop-up</i> Berhasil <i>Update Halaman User</i> <i>Manual</i>	92
Gambar 55. Implementasi <i>History</i> Halaman <i>User Manual</i>	92
Gambar 56. Implementasi Tampilan <i>Pop-up Delete History</i> Halaman <i>User Manual</i>	93
Gambar 57. Implementasi Tampilan <i>Pop-up</i> Berhasil Hapus <i>History</i> Halaman <i>User</i> <i>Manual</i>	93
Gambar 58. Implementasi Tampilan <i>Pop-up Delete User Manual</i> Aplikasi.....	94
Gambar 59. Implementasi Tampilan <i>Pop-up</i> Berhasil Hapus <i>User Manual</i> Aplikasi	95

Gambar 60. Implementasi Halaman <i>What's New</i> Pengguna.....	95
Gambar 61. Implementasi Halaman Detail <i>What's New</i> Pengguna	96
Gambar 62. Implementasi Halaman <i>What's New</i> Admin.....	96
Gambar 63. Implementasi Halaman ‘Buat’ <i>Whats New</i>	97
Gambar 64. Implementasi Tampilan <i>Pop-up</i> Berhasil Membuat <i>What's New</i>	98
Gambar 65. Implementasi Halaman <i>History What's New</i>	99
Gambar 66. Implementasi Halaman Detail <i>History What's New</i>	99
Gambar 67. Implementasi Tampilan <i>Pop-up Delete History What's New</i>	100
Gambar 68. Implementasi Tampilan <i>Pop-up</i> Berhasil Hapus <i>History What's New</i>	100
Gambar 69. Implementasi Halaman Detail <i>What's New</i> Admin.....	101
Gambar 70. Implementasi Halaman <i>Edit What's New</i>	102
Gambar 71. Implementasi Tampilan <i>Pop-up</i> Berhasil <i>Update Data What's New</i>	103
Gambar 72. Implementasi Tampilan <i>Pop-up Delete What's New</i>	103
Gambar 73. Implementasi Tampilan <i>Pop-up</i> Berhasil Hapus <i>What's New</i>	104
Gambar 74. Implementasi <i>Login Page</i>	105
Gambar 75. Implementasi <i>Login Page</i>	105
Gambar 76. Implementasi Halaman <i>Register</i>	107
Gambar 77. Implementasi Aktivasi Akun.....	108
Gambar 78. Implementasi Halaman <i>Lupa Password</i>	109
Gambar 79. Implementasi Halaman <i>Reset Password</i>	110
Gambar 80. Implementasi Fitur <i>Search</i>	111
Gambar 81. Implementasi Fitur <i>Filter User Manual</i>	112
Gambar 82. Implementasi Pengembangan <i>Chatbot</i> pada <i>Dialogflow Essentials</i>	113
Gambar 83. Integrasi <i>Chatbot</i> ke <i>Website</i>	114
Gambar 84. Implementasi Halaman <i>User Manual</i> Pengguna.....	115
Gambar 85. Implementasi Halaman <i>User Manual</i> Pengguna.....	116
Gambar 86. Implementasi <i>Pop-up</i> Deskripsi Aplikasi	117
Gambar 87. Implementasi <i>Form</i> Buat <i>User Manual</i>	118
Gambar 88. Lanjutan Kode Implementasi <i>Form</i> Buat <i>User Manual</i>	120
Gambar 89. Lanjutan Kode Implementasi <i>Form</i> Buat <i>User Manual</i>	121

Gambar 90. Implementasi Kode Halaman <i>History</i>	122
Gambar 91. Implementasi Tampilan <i>Pop-up</i> Deskripsi Aplikasi Halaman Admin	123
Gambar 92. Implementasi Tampilan <i>Pop-up Save</i> atau <i>Edit</i> Deskripsi Aplikasi	124
Gambar 93. Implementasi <i>Form Edit User Manual</i>	125
Gambar 94. Implementasi <i>Form Edit User Manual</i>	126
Gambar 95. Implementasi <i>Delete User Manual</i>	127
Gambar 96. Implementasi Halaman <i>What's New</i> Pengguna	128
Gambar 97. Implementasi Halaman Detail <i>What's New</i> Pengguna	128
Gambar 98. Implementasi Halaman 'Buat' <i>What's New</i>	129
Gambar 99. Implementasi Halaman <i>History What's New</i>	130
Gambar 100. Implementasi Halaman <i>Edit What's New</i>	131
Gambar 101. Implementasi Halaman <i>Edit What's New</i>	132
Gambar 102. Implementasi Tampilan <i>Pop-up Delete What's New</i>	133
Gambar 103. Kesalahan Konfigurasi <i>Chatbot</i>	140
Gambar 104. Potongan Kode 'AuthController.php' sebelum Dilakukan Perbaikan	141
Gambar 105. Potongan Kode 'AuthController.php' setelah Dilakukan Perbaikan	142
Gambar 106. Potongan Kode 'Routes.php' sebelum Dilakukan Perbaikan	143
Gambar 107. Potongan Kode 'Routes.php' setelah Dilakukan Perbaikan	143
Gambar 108. <i>Bug</i> pada Halaman <i>User Manual</i>	144
Gambar 109. Potongan Kode Detail.php Sebelum Perbaikan	145
Gambar 110. Potongan Kode Detail.php Setelah Dilakukan Perbaikan	146
Gambar 111. Arsitektur <i>Deployment</i>	147
Gambar 112. Proses <i>Deployment</i>	148

BAB I. PENDAHULUAN

1.1 Latar Belakang

Knowledge base adalah sistem pakar yang memodelkan pengetahuan milik manusia sebagai seperangkat aturan yang bersifat kolektif. Sistem ini dapat berisi seperangkat aturan dalam suatu sistem perangkat lunak yang dapat digunakan oleh orang lain dalam organisasi, contohnya seperti *user manual* [1]. *User manual* merupakan komponen yang harus ada dalam sebuah proyek pengembangan *software* pada saat produk aplikasi akan dirilis dan didistribusikan untuk digunakan [2]. *User manual* di BPJS Ketenagakerjaan umumnya masih berbentuk *file* digital, seperti *file* dengan ekstensi PDF atau Word. Hal tersebut menimbulkan kesulitan pembaruan informasi, penumpukan *file* dan informasi jika terjadi perubahan, dan akibat negatif lainnya yang merugikan perusahaan.

Pengembangan *knowledge base* berbasis *website* merupakan solusi untuk menyelesaikan permasalahan yang dihadapi. *Knowledge base* berbasis *website* memungkinkan pengelolaan dan pembaruan informasi secara langsung pada *website* yang dapat dilakukan oleh admin dengan meng-*input*-kan atau meng-*upload* informasi sehingga tidak perlu membuat *file user manual* yang baru [3]. Selain itu, terdapat fitur baru yang dikembangkan, yaitu *chatbot* yang dapat digunakan oleh pengguna untuk menemukan informasi dengan lebih mudah dan cepat. *Chatbot* memungkinkan pengguna dapat berinteraksi dengan sistem secara *real-time* seperti berkomunikasi dengan manusia [4].

Perkembangan teknologi informasi mengakibatkan banyak perusahaan beralih ke *platform* yang lebih interaktif dan menarik. Pengembangan *knowledge base* berbasis *website* serta *chatbot* berkontribusi dalam pelatihan dan pengembangan karyawan yang lebih baik. Ketika pengguna menemukan kendala dalam penggunaan aplikasi, pengguna dapat mengakses *knowledge base* berbasis

website untuk mencari informasi yang diperlukan. Pengembangan *knowledge base* dari *file* digital menjadi *website* relevan untuk memenuhi kebutuhan pengguna mengenai aplikasi maupun layanan BPJS Ketenagakerjaan [5].

Knowledge base berbasis *website* dan pengembangan fitur baru, yaitu *chatbot* diharapkan dapat mempercepat proses pelatihan karyawan, memastikan karyawan dan pemangku kepentingan lainnya mendapat informasi yang akurat dan terkini, serta mengurangi kesalahan dalam penggunaan aplikasi BPJS Ketenagakerjaan. Selain itu, sistem ini diharapkan dapat memberikan fleksibilitas bagi BPJS Ketenagakerjaan untuk melakukan perubahan informasi tanpa memerlukan biaya tambahan yang signifikan dibandingkan dengan sebelum adanya sistem ini.

1.2 Tujuan

Tujuan dari pengembangan *knowledge base* adalah sebagai berikut :

1. Mengukur kemudahan pengguna dalam menemukan informasi terkait penggunaan aplikasi yang digunakan oleh karyawan BPJS Ketenagakerjaan.
2. Memastikan pengembangan *knowledge base* berbasis *website* yang dikembangkan sesuai dengan kebutuhan pengguna.

1.3 Rumusan Masalah

Adapun rumusan masalah dari pengembangan *knowledge base* ini adalah sebagai berikut :

Bagaimana proses pengembangan *knowledge base* berbasis *website* yang dilakukan untuk memenuhi kebutuhan pengguna di BPJS Ketenagakerjaan?

1.4 Manfaat Penelitian

Adapun manfaat penelitian dari pengembangan *knowledge base* adalah sebagai berikut :

1. Meningkatkan pengalaman pengguna dalam mengakses informasi panduan pengguna dalam menggunakan aplikasi BPJS Ketenagakerjaan.
2. Meningkatkan kemudahan dalam mengakses informasi sehingga pengguna dapat menemukan informasi yang dibutuhkan dengan lebih cepat sehingga dapat menghemat waktu dan tenaga.
3. Penelitian memberikan wawasan baru untuk mengidentifikasi perbaikan untuk pengembangan selanjutnya.

1.5 Batasan Masalah

Adapun batasan masalah dari penelitian ini adalah sebagai berikut :

1. Data *user manual* yang dapat digunakan hanya data *user manual* aplikasi eksternal BPJS Ketenagakerjaan yang dapat diakses oleh publik.
2. *User manual* berbasis *website* ini belum menggunakan media interaktif (multimedia).
3. Karyawan BPJS Ketenagakerjaan hanya dapat mengakses informasi yang sudah disediakan dalam *website*, sedangkan admin dapat melakukan pengelolaan *knowledge base*, yaitu menambahkan, meng-*update*, dan menghapus *user manual*.
4. Sistem keamanan menggunakan sistem keamanan *default* yang disediakan oleh *framework* yang digunakan, yaitu *framework* Codeigniter 4.
5. Penelitian ini dilakukan dalam periode waktu delapan bulan dan dilakukan pada karyawan BPJS Ketenagakerjaan sebagai pengguna *knowledge base* berbasis *website* tersebut.

1.6 Sistematika Penulisan

Adapun sistematika penulisan yang digunakan adalah sebagai berikut :

BAB I Pendahuluan

Bab ini menjelaskan mengenai latar belakang, tujuan, rumusan masalah, manfaat penelitian, Batasan masalah, dan sistematika penulisan.

BAB II Tinjauan Pustaka

Bab ini tinjauan terkait dengan obyek penelitian, metode atau pun alat yang digunakan pada penelitian, dan penelitian terkait.

BAB III Metode Penelitian

Bab ini menjelaskan mengenai waktu, tempat, alat, bahan, dan tahapan penelitian yang dilakukan.

BAB IV Hasil dan Pembahasan

Bab ini menjelaskan hasil analisis dan pembahasan dari pelaksanaan penelitian yang telah dilakukan.

BAB V Kesimpulan dan Saran

Bab ini adalah penutup yang berisi kesimpulan dari hasil analisis dan saran berdasarkan penelitian yang telah dilakukan.

BAB II. TINJAUAN PUSTAKA

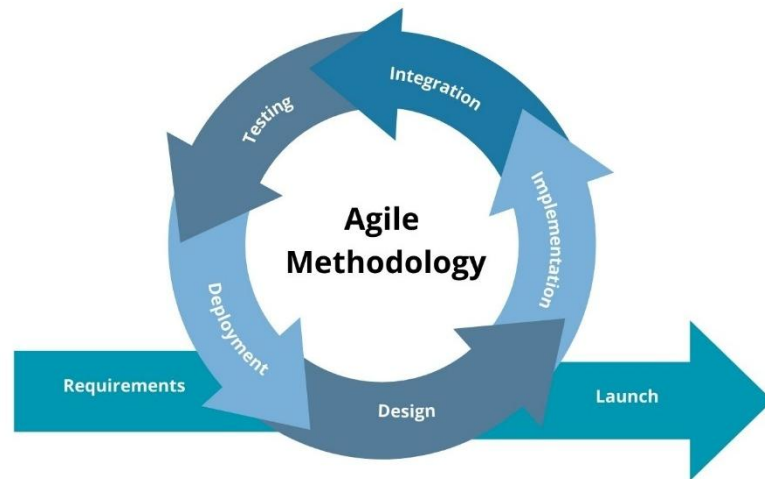
2.1 *Knowledge Base*

Sistem pakar adalah teknik yang digunakan untuk menangkap domain keahlian manusia yang sangat spesifik dan terbatas berdasarkan prinsip dasar seorang ahli. Sistem pakar memodelkan pengetahuan manusia sebagai seperangkat aturan secara kolektif yang disebut dengan *knowledge base*. Sistem ini dapat berisi pengetahuan karyawan yang terampil dalam bentuk seperangkat aturan dalam suatu sistem perangkat lunak yang bisa digunakan oleh orang lain dalam organisasi. Seperangkat aturan tersebut dapat disimpan dan menambah memori atau pembelajaran pada perusahaan. Pengetahuan yang dimiliki manusia harus dimodelkan atau direpresentasikan sesuai dengan bagaimana komputer dapat memproses. Sistem pakar memiliki 200 hingga ribuan aturan yang bergantung pada kompleksitas masalahnya [1].

Knowledge base dapat digambarkan sebagai tempat penyimpanan informasi yang terpusat. *Knowledge base* yang efektif dapat memberikan informasi yang tepat dengan sedikit usaha sehingga membantu untuk memberikan layanan yang lebih baik. Sistem ini memungkinkan pengguna untuk menemukan informasi yang relevan dengan mudah dan cepat. *Knowledge base* sering kali digunakan dalam konteks layanan pelanggan, *e-learning*, dan manajemen informasi. *Knowledge base* memiliki beberapa fungsi penting, yaitu sebagai penyimpanan berbagai jenis informasi dalam format terstruktur, seperti artikel, FAQ (*Frequently Asked Question*), dan panduan pengguna. Sistem ini dibuat untuk mengurangi waktu dan biaya dalam memberikan informasi kepada pengguna karena pengguna dapat menemukan jawaban secara mandiri ketika menemukan kendala. Komponen utama dari sistem ini terdiri dari *database* untuk menyimpan data dan informasi dan

antarmuka pengguna untuk memungkinkan pengguna berinteraksi dengan sistem. [6].

2.2 Metode *Agile*



Gambar 1. Tahapan Metode *Agile* [7]

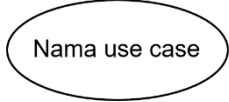
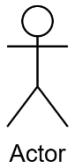
Metode *Agile* adalah pendekatan pengembangan perangkat lunak yang berfokus pada kolaborasi, pengembangan bersama pengguna, dan pengembangan singkat dengan iterasi yang singkat [8]. Penerapan metode *Agile* dalam pengembangan perangkat lunak meliputi beberapa alur penelitian, yaitu tahap *requirements*, *design*, *implementation*, *integration*, *testing*, dan *deployment* pada setiap proses iterasi. Pada tahap *requirements* dilakukan studi dan analisis tentang kebutuhan pengguna agar sistem yang dibuat sesuai dengan kebutuhan pengguna. Pada tahap *design* dilakukan perancangan sistem untuk memberikan rencana solusi yang dilakukan dari masalah yang dihadapi oleh pengguna. Tahap *implementation* merupakan tahap mengimplementasikan rencana yang sudah disepakati ke dalam desain dan juga coding. Tahap *integration* merupakan tahap penggabungan kode dan mengintegrasikan *frontend* dan *backend* dari sistem. Tahap *testing* dilakukan pengujian sistem yang sudah dibuat untuk memastikan sistem yang dibuat sudah sesuai dengan rencana yang telah dibuat dan disepakati. Tahap *deployment* merupakan tahap penyebaran sistem atau aplikasi ke lingkungan produksi. Jika sistem memerlukan perbaikan dan periode sistem sudah habis, maka dilakukan

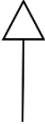
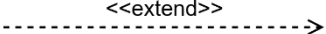
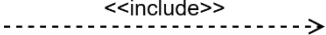
iterasi dengan memperbarui *requirements* berdasarkan umpan balik dari pengguna[7].

2.3 UML (*Unified Modelling Language*)

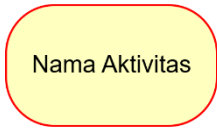
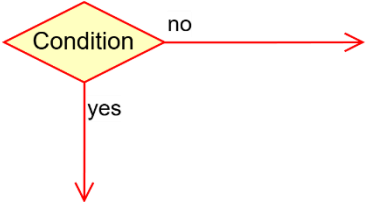
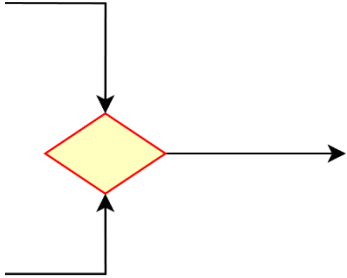
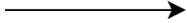
UML adalah diagram yang dibuat untuk memodelkan aplikasi dan sistem perangkat lunak. Diagram ini dibuat menggunakan mekanisme yang telah disediakan. Terdapat beberapa diagram yang dapat digunakan untuk memodelkan aplikasi dan sistem yang dikembangkan, contohnya adalah *use case diagram* dan *activity diagram*. *Use case diagram* adalah diagram yang digunakan untuk memodelkan fungsi atau proses dari sebuah aplikasi atau sistem dan menjelaskan mengenai hubungan antara fungsi-fungsinya dengan entitas eksternal yang disebut aktor. *Activity diagram* adalah diagram yang digunakan untuk memodelkan aktivitas yang saling berhubungan melalui alur kerja dan menunjukkan urutan, pengecualian, sinkronisasi, serta hubungan konkuren. Berikut merupakan tabel yang menjelaskan komponen yang berada di dalam *use case diagram* dan *activity diagram* [7].

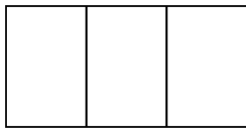
Tabel 1. *Use Case Diagram* [7]

Nama	Deskripsi	Notasi
<i>Use case</i>	<i>Use case</i> adalah proses bisnis yang diberi nama, dimulai dengan seorang <i>actor</i> dan diakhiri dengan <i>actor</i> tersebut, serta menyelesaikan tugas bisnis untuk <i>actor</i> tersebut	
<i>Actor</i>	<i>Actor</i> adalah peran yang dipakai sekumpulan entitas bisnis atau pemangku kepentingan yang berada di luar sistem dan berinteraksi dengan sistem tersebut.	

<i>Association</i>	<i>Association</i> diantara <i>actor</i> dan <i>use case</i> menunjukkan bahwa <i>actor</i> tersebut menggunakan <i>use case</i> tersebut.	
<i>Inheritance</i>	<i>Inheritance</i> adalah hubungan biner antara dua konsep di mana satu konsep adalah generalisasi atau spesialisasi dari yang lainnya.	
<i>Extension</i>	<i>Extension</i> diantara <i>actor</i> dan <i>use case</i> adalah sebuah hubungan biner antara dua <i>use case</i> di mana satu <i>use case</i> dapat melanjutkan proses bisnis dari yang lainnya.	
<i>Inclusion</i>	<i>Inclusion</i> diantara <i>actor</i> dan <i>use case</i> adalah sebuah hubungan biner antara dua <i>use case</i> di mana satu <i>use case</i> mencakup yang lainnya sebagai bagian dari proses bisnisnya.	

Tabel 2. Activity Diagram [7]

Nama	Deskripsi	Notasi
<i>Activity / action</i>	<i>Activity</i> atau <i>action</i> adalah sebuah aktivitas komputasi atau pemrosesan informasi.	
<i>Conditional branching</i>	<i>Conditional branching</i> digunakan untuk memilih salah satu dari sejumlah alternatif jalur.	
<i>Merging alternative threads</i>	<i>Merging alternatives threads</i> digunakan untuk mendefinisikan satu titik keluar untuk jalur alternatif yang dibuat oleh percabangan bersyarat.	
<i>Control flow</i>	<i>Control flow</i> digunakan untuk mendefinisikan hubungan prioritas antara dua aktivitas.	
<i>Initial node</i>	<i>Initial node</i> adalah sebuah titik awal untuk memulai proses atau alur kerja.	
<i>Final node</i>	<i>Final node</i> adalah titik akhir untuk mengakhiri proses atau alur kerja.	

<i>Swin lanes</i>	<i>Swim lanes</i> adalah mekanisme untuk mengelompokkan atau mengatur aktivitas sesuai dengan organisasi atau subsistem.	
-------------------	--	---

2.4 Skala Likert

Skala Likert memiliki kegunaan untuk mengumpulkan data mengenai sikap dan pendapat mengenai suatu hal. Skala ini digunakan sebagai alat ukur untuk menilai persepsi responden terhadap suatu pernyataan atau item. Penilaian ini biasanya terdiri dari serangkaian pernyataan beserta pilihan jawaban yang menunjukkan tingkat persetujuan atau ketidaksetujuan responden terhadap suatu hal. Skala Likert umumnya memiliki 5 atau 7 pilihan jawaban yang terdiri dari kategori, seperti sangat tidak setuju, tidak setuju, netral, setuju, sangat setuju. Kelebihan dari skala ini adalah memudahkan pengumpulan data dari banyak responden dan memberikan data yang dapat dianalisis secara statistik. Kekurangan dari skala ini adalah interpretasi kategori dapat bervariasi antar individu yang dapat menyebabkan ketidakseimbangan dalam data [9].

2.5 UAT (*User Acceptance Testing*)

UAT adalah tahap akhir dalam proses pengembangan perangkat lunak dimana pengguna akhir menguji sistem untuk memastikan bahwa perangkat lunak memenuhi kebutuhan dan harapan pengguna sebelum melakukan tahap *deployment*. UAT bertujuan untuk mengidentifikasi masalah atau kekurangan yang mungkin tidak terdeteksi selama tahap pengujian sebelumnya. UAT membantu dalam mengevaluasi kinerja sistem dan kemudahan pengguna dalam menggunakan sistem. Jika hasil pengujian positif, maka pengembangan dapat dilanjutkan ke tahap *deployment*. Akan tetapi, jika terdapat revisi, maka pengembangan kembali ke siklus pengembangan untuk melakukan perbaikan [8]. Berikut merupakan rumus

yang digunakan untuk menghitung skor aktual hasil pengujian menggunakan *User Acceptance Testing*.

$$\% \text{ Skor aktual} = \frac{\text{Skor Aktual}}{\text{Skor Ideal}} \times 100 \%$$

Keterangan :

- Skor aktual adalah hasil respons dari keseluruhan responden kuesioner.
- Skor ideal adalah skor tertinggi yang didapat ketika semua responden memilih skor tertinggi untuk setiap jawaban [10].

2.6 *Chatbot*

Chatbot adalah program komputer yang dibuat untuk membantu percakapan manusia dengan sistem dengan mensimulasikannya secara *real time* melalui teks atau suara. Pengembangan sistem ini memungkinkan manusia berkomunikasi dengan komputer layaknya sedang berkomunikasi dengan manusia. *Chatbot* dirancang untuk melakukan interaksi yang spesifik dan terstruktur. *Chatbot* dapat diaplikasikan di berbagai *platform* yang membutuhkannya, seperti *website*, aplikasi, dan media sosial [4]. *Chatbot* yang dibuat bergantung pada *data train* yang ditentukan oleh pengembang *chatbot* saat mengenali dan merespons dengan metode penandaan untuk setiap pola kalimat untuk meningkatkan akurasi pada sistem yang dikembangkan [11].

2.7 Aplikasi yang Digunakan

2.7.1 Dialogflow Essentials

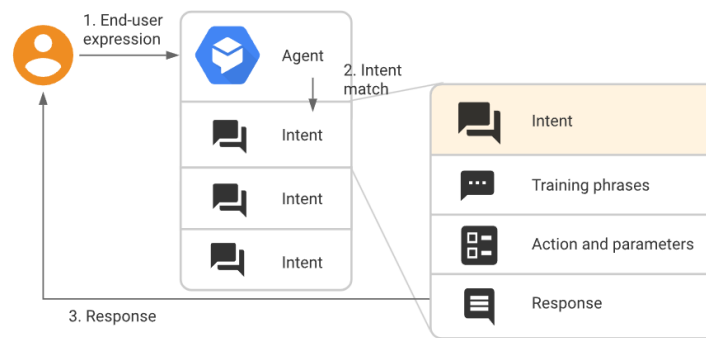
Dialogflow Essentials (ES) adalah salah satu aplikasi pengembangan *chatbot* dan antarmuka percakapan berbasis *Natural Language Processing* (NLP) yang dikembangkan oleh Google Cloud. NLP merupakan bagian dari kecerdasan buatan yang berkaitan dengan interaksi antara komputer dan manusia melalui bahasa alami. Tujuan utama dari NLP adalah memungkinkan komputer untuk menafsirkan, memahami, dan menghasilkan bahasa manusia secara bermakna dan kontekstual. Konsep tersebut yang digunakan oleh Dialogflow ES sehingga aplikasi ini memungkinkan pengembang untuk membangun *chatbot* yang dapat memahami

maksud pengguna melalui teks serta memberikan respons yang sesuai berdasarkan *intent* yang telah didefinisikan.

Dialogflow ES menyediakan antarmuka berbasis *web* yang intuitif untuk mendesain, mengatur, dan mengelola komponen *chatbot* tanpa memerlukan keahlian pemrograman yang kompleks. Beberapa fitur pada Dialogflow ES yang menerapkan NLP adalah *intent matching*, *entity recognition*, *training phrase generalization*, dan *fallback handling*. *Intent matching* digunakan untuk memahami maksud kalimat yang diinputkan pengguna dan *entity recognition* digunakan untuk mengambil informasi penting dari kalimat. Fitur *training phrase generalization* digunakan untuk memahami variasi bahasa dari pengguna dan *fallback handling* digunakan untuk memberikan tanggapan saat maksud pengguna tidak dipahami.

Pada Dialogflow ES terdapat beberapa komponen yang dapat digunakan untuk membangun *chatbot*. *Agent* pada Dialogflow ES adalah modul natural language understanding yang menangani percakapan serentak dengan pengguna akhir. Dialogflow menerjemahkan teks dari pengguna akhir selama percakapan ke data struktur yang dapat dipahami oleh aplikasi. *Intent* pada Dialogflow ES digunakan untuk mengategorikan niat pengguna akhir untuk satu giliran percakapan. Dialogflow ES akan mencocokkan input pengguna dengan *intent* terbaik pada *agent* untuk memberikan respons yang sesuai.

Pada *intent* terdapat beberapa komponen yang berperan dalam proses pencocokkan *intent* untuk memberikan respons kepada pengguna, yaitu *training phrases*, *action and parameters*, dan *response*. Diagram di bawah ini menunjukkan alur prosesnya.



Gambar 2. Alur Dasar Pencocokan *Intent* untuk Memberikan Respons ke Pengguna Akhir

Proses yang terjadi adalah pengguna mengirimkan pernyataan atau pertanyaan dalam bentuk teks ke sistem *chatbot* yang dibangun menggunakan Dialogflow ES. Kemudian agent menerima input tersebut dan menganalisis serta mencocokkan maksud dari input pengguna dengan salah satu *intent* yang telah didefinisikan oleh pengembang. Proses pencocokan ini menyesuaikan dengan *training phrases* yang telah dibuat pada masing-masing *intent*. Setelah *intent* dikenali, Dialogflow ES dapat mengekstrak parameter (*entity*) dari *input* yang digunakan untuk memicu aksi tertentu, seperti pemanggilan API atau logika sistem. Setelah *intent* dikenali dan parameter diekstrak, Dialogflow ES akan memberikan respons yang sesuai berdasarkan konfigurasi di dalam *intent* tersebut.

Selain itu, Dialogflow ES juga mendukung integrasi langsung dengan berbagai *platform* populer seperti Google Assistant, Facebook Messenger, Telegram, dan aplikasi berbasis *web*. Pengembang dapat menggunakan salah satu dari opsi tersebut untuk proses integrasi. Interaksi pengguna akhir akan langsung ditangani oleh sistem sehingga pengembang dapat berfokus untuk membuat agen. Pengembang dapat melihat dokumentasinya untuk masing-masing platform integrasi untuk mengetahui detail penggunaannya [12].

2.7.2 Visual Studio Code

Visual Studio Code (VS Code) adalah editor kode sumber yang dikembangkan oleh Microsoft dengan antarmuka pengguna yang intuitif. Aplikasi

ini dapat digunakan untuk berbagai bahasa pemrograman dan integrasi dengan berbagai alat pengembangan sehingga mempermudah pengembangan aplikasi *web*, *mobile*, dan *cloud*. VS Code dapat dijalankan di berbagai sistem operasi, seperti Windows, macOS, dan Linux. Pengguna juga dapat menambahkan fungsionalitas tambahan melalui *marketplace* dan aplikasi ini memudahkan pengelolaan repositori langsung dari *editor* dengan integrasi Git. Selain itu, VS Code memiliki fitur *extensions* yang membantu dan mempermudah pengguna dalam setiap proses pengembangan[13].

2.7.3 MySQL

MySQL adalah salah satu sistem manajemen basis data SQL yang populer digunakan. Perangkat lunak MySQL menggunakan lisensi GPL (*GNU General Public*) dan merupakan perangkat lunak yang bersifat *open source*. Sistem ini dikembangkan, didistribusikan, dan didukung oleh perusahaan Oracle. MySQL memiliki fungsi untuk mengelola kumpulan data terstruktur yang dibutuhkan dalam proses pengembangan. Basis data MySQL membantu dalam menambah, mengakses, dan memproses data yang disimpan dalam basis data. Sistem ini menyimpan data dalam tabel terpisah dan disusun menjadi *file* fisik yang dioptimalkan untuk kecepatan dan fleksibel [14].

2.7.4 phpMyAdmin

phpMyAdmin adalah alat perangkat lunak gratis yang dibuat dalam bahasa PHP dan dirancang untuk membantu administrasi MySQL. Aplikasi ini dapat diakses melalui *web* dan menyediakan antarmuka yang intuitif yang membantu dalam pengelolaan basis data MySQL. Aplikasi ini memungkinkan pengguna untuk melakukan administrasi basis data tanpa menulis perintah SQL secara manual. phpMyAdmin menyediakan operasi untuk mengelola basis data, tabel, kolom, relasi, indeks, pengguna, dan hak akses. Pengguna juga dapat melakukan

pengelolaan struktur basis data dengan melakukan penelusuran, pembuatan, penyalinan, penghapusan, penggantian nama, dan pengubahan basis data, tabel, *view*, kolom, serta indeks. Selain itu, phpMyAdmin mendukung pengelolaan akun pengguna MySQL dan mengatur hak istimewa (*privileges*) pada basis data atau tabel tertentu. Aplikasi ini mendukung impor data dari berbagai format file ke dalam basis data dan ekspor data dari basis data ke berbagai format yang berbeda yang sangat membantu untuk backup data dan migrasi data [15].

2.7.5 XAMPP

XAMPP adalah salah satu *web server* yang digunakan untuk melakukan pengujian aplikasi yang sedang dikembangkan. *Web server* ini adalah *server* yang paling populer digunakan oleh *programmer*. Kelebihan dari XAMPP adalah fiturnya yang lengkap dan sangat mudah untuk digunakan untuk pengujian aplikasi secara lokal. Selain itu, aplikasi dapat diunduh dengan mudah dengan mengunjungi *website* resmi XAMPP. Setelah proses pengunduhan sudah dilakukan, pengguna dapat langsung melakukan instalasi XAMPP langsung di komputer pengguna [16].

2.7.6 draw.io

draw.io adalah salah satu alat *diagramming open-source* yang dapat diakses dan digunakan melalui *web browser*. Aplikasi ini menyediakan *platform* yang mudah digunakan untuk membuat berbagai jenis diagram, mulai dari diagram alir, diagram UML, *mind maps*, diagram jaringan, *mockup*, dan lain-lain. draw.io menyediakan pustaka *shape* (bentuk) dan *template* yang beragam untuk membantu dalam proses pembuatan diagram. Aplikasi ini juga memiliki kemampuan untuk berintegrasi dengan berbagai layanan penyimpanan seperti Google Drive, OneDrive, Dropbox, dan Github. Selain itu, diagram yang dibuat di aplikasi ini dapat secara mudah untuk diekspor ke dalam berbagai format gambar dan dokumen, seperti PNG, JPEG, SVG, PDF, dan XML. Antarmuka aplikasi ini dirancang untuk mudah dipelajari dan digunakan dengan fitur *drag-and-drop* yang sangat membantu untuk membuat diagram dengan cepat dan efisien [17].

2.7.7 dbdiagram.io

dbdiagram.io adalah alat diagramming basis data secara online yang dirancang untuk membantu dalam merancang dan memvisualisasikan basis data dalam bentuk Entity Relationship Diagram (ERD) dengan cepat dan mudah. Aplikasi ini dirancang dengan kemudahan dalam mendefinisikan struktur basis data menggunakan sintaks teks deklaratif yang intuitif yang dapat dikonversi menjadi diagram visual secara otomatis. Aplikasi ini juga mendukung berbagai jenis relasi antar tabel seperti one-to-one, one-to-many, dan many-to-many yang divisualisasikan dalam ERD yang dihasilkan. dbdiagram.io menyediakan fitur ekspor yang beragam untuk mengeksport diagram dalam berbagai format, seperti PNG, PDF, atau format DBML itu sendiri untuk dibagikan dan diedit ulang. dbdiagram.io juga menyediakan kemampuan untuk mengimpor skema basis data yang sudah ada dari basis data SQL [18].

2.7.8 Figma

Figma adalah alat desain berbasis *web* yang memungkinkan pengguna untuk membuat *prototype* dan desain antarmuka pengguna secara kolaboratif. Figma dikenal karena kemudahan akses dan kemampuan kolaborasi secara *real-time*. Aplikasi ini dapat diakses dari berbagai perangkat tanpa perlu instalasi dan dapat terintegrasi dengan berbagai alat kolaborasi dan manajemen proyek. Aplikasi ini telah menjadi alat yang sering digunakan oleh UI/UX *designer* untuk menciptakan *wireframe*, *prototype*, dan desain akhir. Adanya fitur kolaborasi memudahkan tim dalam memberikan umpan balik dan melakukan iterasi desain jika terdapat perubahan [19].

2.7.9 Bootstrap 4

Bootstrap 4 adalah *framework frontend* yang dirancang untuk memudahkan pengembangan situs *web* responsif dan *mobile-first*. *Framework* yang dikembangkan oleh Twitter ini menyediakan kumpulan alat CSS dan JavaScript yang memungkinkan pengguna pengembang untuk membuat antarmuka yang menarik dan fungsional dengan cepat. Beberapa fitur utama yang dapat digunakan adalah *grid system*, komponen UI, dan *utility classes*. *Framework* ini dapat

mempercepat proses pengembangan antarmuka pengguna dengan komponen dan *grid* yang sudah tersedia [20].

2.7.10 IDCloudHost

IDCloudHost adalah penyedia layanan *web hosting* yang berasal dari Indonesia. Layanan *web hosting* ini menyediakan berbagai jenis produk, yaitu *shared hosting*, *Virtual Private Server (VPS)*, *cloud hosting*, *colocation server*, serta layanan registrasi domain. Penyedia layanan ini menawarkan keunggulan dalam kecepatan akses dan stabilitas koneksi bagi pengguna di wilayah Indonesia sehingga cocok untuk digunakan dalam pengembangan *website* dengan target pengguna lokal. IDCloudHost menyediakan beberapa fitur yang dapat membantu, seperti SSL gratis, *control panel* (cPanel atau DirectAdmin), serta layanan *cloud VPS* yang dapat dikustomisasi sesuai dengan kebutuhan *developer*. Selain itu, IDCloudHost juga menyediakan fasilitas *cloud object storage*, *email hosting*, dan layanan keamanan tambahan seperti *cloudflare protection*. IDCloudHost digunakan untuk menempatkan *file website* dan *database* supaya dapat diakses secara daring[21].

2.7.11 Codeigniter 4

Codeigniter 4 adalah *framework* PHP yang dirancang untuk pengembangan aplikasi *web* yang cepat dan efisien. *Framework* ini menyediakan arsitektur MVC (*Model, View, Controller*) yang memisahkan logika aplikasi dari tampilan sehingga memudahkan pengembangan dan pemeliharaan aplikasi. Konsep MVC membagi tugas aplikasi menjadi tiga bagian utama, yaitu *controller* bertugas mengatur alur logika dan menerima *request* dari user, lalu memanggil *model* untuk mengelola data ke *database*, dan akhirnya mengirim data ke *view* untuk ditampilkan ke pengguna dalam bentuk halaman *web*. *Model* berfungsi untuk menangani semua operasi *database* seperti mengambil, menambah, mengubah, atau menghapus data. *View* adalah bagian yang mengatur tampilan antarmuka pengguna menggunakan HTML, CSS, dan PHP. Sementara itu, routes mengatur URL yang diakses pengguna agar diarahkan ke *controller* dan metode yang sesuai. Sistem ini membuat struktur kode lebih rapi, teratur, dan mudah dikembangkan. Selain itu, Codeigniter 4

menyediakan berbagai metode untuk berinteraksi dengan *database*, termasuk *active record* dan *query builder* [22].

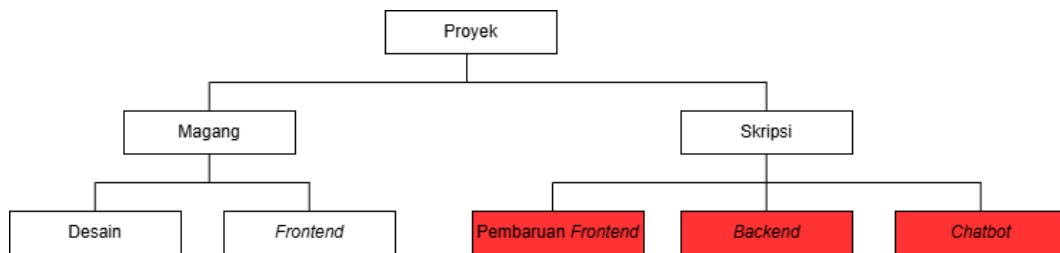
2.7.12 Postman

Postman adalah sebuah *platform* yang digunakan untuk mengembangkan, menguji, dan mendokumentasikan API. Postman mempermudah para *developer* untuk melakukan simulasi permintaan ke API dan memeriksa respons yang diberikan oleh *server* tanpa perlu membuat antarmuka pengguna terlebih dahulu. Postman mendukung berbagai jenis metode HTTP, seperti GET untuk mengambil data, POST untuk mengirim data, PUT untuk memperbarui data, dan DELETE untuk menghapus data. Beberapa fitur utama Postman sangat berguna dalam proses pengembangan *request builder*, *environment variables*, *collections*, *test scripts*, serta *monitoring* dan dokumentasi. Postman digunakan dalam pengembangan *web* untuk menguji *endpoint* API yang dikembangkan di *backend*, menverifikasi integrasi antara *frontend* dan *backend* sebelum implementasi UI, dan menganalisis data JSON atau XML yang diterima dari *server*[23].

2.7.13 API (Application Programming Interface)

Application Programming Interface adalah sekumpulan aturan, protokol, dan alat yang memungkinkan satu perangkat lunak untuk berkomunikasi dengan perangkat lunak lainnya. API berfungsi sebagai jembatan yang memungkinkan integrasi dan pertukaran data antar berbagai sistem, layanan, atau aplikasi dengan cara yang terstruktur dan aman. API dalam konteks pengembangan aplikasi *web* digunakan untuk mengakses data dari sumber eksternal, menghubungkan *backend* dan *frontend* pada aplikasi, mengintegrasikan layanan pihak ketiga seperti Google Maps, pembayaran *online*, dan media sosial. Salah satu jenis API yang paling umum digunakan adalah REST API (*Representational State Transfer*) yang menggunakan protokol HTTP untuk mengakses dan memanipulasi data dalam format JSON atau XML. API dalam aplikasi berbasis *website* digunakan untuk menampilkan data dari *database* eksternal tanpa perlu menyimpan salinan lokal dan menghubungkan aplikasi *framework* yang digunakan dengan sistem manajemen file[23].

2.8 Sistem Informasi



Gambar 3 Bagan Proyek

Gambar 3 merupakan bagan yang menjelaskan proyek yang telah dikerjakan selama magang dan dikerjakan untuk skripsi. Pada saat magang telah dilaksanakan proses desain *website* yang dikembangkan dan telah diimplementasikan dalam *frontend*. Proyek tersebut dilakukan bersama dengan tim yang merupakan peserta magang lainnya. Desain yang telah dibuat, yaitu desain yang telah disetujui oleh *stakeholder* dijadikan referensi untuk proses pengembangan selanjutnya. Selain itu, telah dilakukan pengembangan *backend* dari sistem. Kemudian, dilakukan beberapa pembaruan *frontend* untuk menyesuaikan kembali dengan kebutuhan pengguna. Setelah dilakukan diskusi dengan *stakeholder* didapatkan bahwa fitur *chatbot* perlu dikembangkan untuk lebih memudahkan pengguna dalam menemukan informasi yang dibutuhkan. Untuk itu, fitur *chatbot* dikembangkan untuk memenuhi kebutuhan pengguna tersebut.

2.9 Penelitian Terdahulu

1.9.1 Pembuatan *User Manual* Berbasis Java sebagai Tutorial Aplikasi Sistem Informasi pada Kader Posyandu Rajawali Singosari

Penelitian yang berjudul “Pembuatan *User Manual* Berbasis Java sebagai Tutorial Aplikasi Sistem Informasi pada Kader Posyandu Rajawali Singosari” ini berfokus pada pengembangan aplikasi *user manual* berbasis Java untuk membantu kader Posyandu dalam memahami aplikasi sistem informasi. Aplikasi ini dibuat untuk menyediakan tutorial berbasis Java yang menjelaskan alur sistem dan kosakata yang sulit dipahami. *User manual* ini berperan penting dalam

mengedukasi kader dalam penggunaan teknologi. Kekurangan dari pengembangan ini adalah *sistem user manual* ini hanya tersedia dalam *platform desktop* yang berarti hanya dapat diakses melalui komputer atau laptop. Selain itu, belum ada informasi yang cukup mengenai uji coba lapangan secara mendalam bersama kader Posyandu [24].

1.9.2 Platform Knowledge-Based Website untuk Meningkatkan Visibilitas UMKM di Sektor Perikanan

Penelitian yang berjudul Platform Knowledge-Based Website untuk Meningkatkan Visibilitas UMKM di Sektor Perikanan ini berfokus pada pengembangan platform website berbasis pengetahuan untuk meningkatkan visibilitas Usaha Mikro, Kecil, dan Menengah (UMKM) di sektor perikanan di Indonesia, terutama pasca-pandemi Covid-19. *Website* ini bertujuan untuk menyediakan informasi yang komprehensif tentang produk perikanan, termasuk kandungan gizi, manfaat, dan cara memilih ikan yang berkualitas. Pendekatan ini diharapkan dapat membantu UMKM dalam memasarkan produk mereka secara *online*, memperluas jangkauan pasar, dan meningkatkan transaksi. Kekurangan dari penelitian ini adalah beberapa mitra mengalami kesulitan dalam menggunakan teknologi sehingga diperlukan pelatihan lebih lanjut dan dukungan teknis. Selain itu, uji coba *website* dilakukan dalam waktu yang singkat dan tidak mencakup semua variabel pengguna yang relevan [25].

1.9.3 Perancangan Sistem Rekomendasi Perbelanjaan Item Kursi Kayu Pada Toko Online Berbasis Website dengan Metode Knowledge Based Recommendation

Penelitian yang berjudul Perancangan Sistem Rekomendasi Perbelanjaan Item Kursi Kayu pada Toko Online Berbasis Website dengan Metode *Knowledge Based Recommendation* ini bertujuan untuk mengembangkan sistem rekomendasi yang dapat membantu pengguna dalam memilih kursi kayu di toko *online* Nuyanu Store. Metode yang digunakan adalah *knowledge based recommendation* yang berfokus pada pemahaman kebutuhan pengguna berdasarkan atribut tertentu seperti jenis kayu, ukuran, warna, dan harga. Adanya sistem ini diharapkan dapat

membantu calon pembeli untuk menemukan produk yang sesuai dengan preferensi mereka. Kekurangan dari penelitian ini adalah penelitian ini hanya mencakup perancangan dan pemodelan, tanpa evaluasi sistem secara langsung di lapangan untuk menilai efektivitasnya dalam meningkatkan pengalaman pengguna. Penelitian ini belum mencakup umpan balik dari pengguna akhir untuk menilai kepraktisan dan kemudahan penggunaan sistem rekomendasi yang dirancang [26].

1.9.4 Rancang Bangun Aplikasi *Knowledge Management* pada Pelayanan Jasa Berbasis Web dengan Metode *Agile Development* (Studi Kasus : PT. Cakrawala Indonesia Sejahtera)

Penelitian yang berjudul Rancang Bangun Aplikasi *Knowledge Management* pada Pelayanan Jasa Berbasis Web dengan Metode *Agile Development* (Studi Kasus : PT. Cakrawala Indonesia Sejahtera) ini bertujuan untuk mengembangkan sistem manajemen pengetahuan berbasis web yang dapat meningkatkan efisiensi dan efektivitas pengelolaan informasi di PT. Cakrawala Indonesia Sejahtera, yang bergerak di bidang penyaluran tenaga kerja ke Jepang. Metode yang digunakan dalam penelitian ini adalah *Agile Development*, khususnya dengan pendekatan *Scrum*, untuk mengoptimalkan proses pengembangan sistem. Kekurangan dari penelitian ini adalah penelitian ini belum mencakup seluruh kebutuhan pengguna. Interaksi lebih lanjut antara pengguna dan admin belum diimplementasikan sepenuhnya [27].

1.9.5 *Knowledge Base System in Practice : Approaches, Application Areas and Limitations*

Disertasi yang berjudul *Knowledge Base Systems in Practice : Approaches, Application Areas and Limitations* ini berfokus pada pengembangan sistem berbasis KBS (Knowledge Base) yang menekankan pentingnya pemisahan antara pengetahuan domain dengan tugas inferensi sesuai dengan paradigma KBP (Knowledge Base Paradigm). Pada penelitian ini, knowledge base dikelompokkan menjadi empat jenis aplikasi, yaitu aplikasi hukum, sistem konfigurasi produk, aplikasi penjadwalan, dan aplikasi pihak ketiga, serta merumuskan metode perolehan pengetahuan yang efektif seperti knowledge articulation dan interactive consultant untuk secara aktif melibatkan pakar domain. Penelitian ini berkaitan

dengan pengembangan sistem knowledge base karena berkaitan dengan pentingnya struktur modular, kejelasan pengetahuan, dan dukungan terhadap logika sistem yang merupakan aspek dalam implementasi sistem berbasis web untuk dokumentasi user manual aplikasi. Di samping itu, terdapat beberapa keterbatasan dalam penelitian ini, yaitu cakupan generalisasi yang terbatas karena pendekatan studi kasus, tidak mencakup semua metode knowledge acquisition yang ada, serta keterbatasan pada jenis aplikasi yang diteliti. Selain itu, banyak implementasi yang masih dalam bentuk prototype sehingga evaluasi jangka Panjang dan skalabilitas belum sepenuhnya dibahas [6].

1.9.6 *Knowledge Based Recommendation Modeling For Clothing Product Selection Recommendation System*

Penelitian yang berjudul *Knowledge Based Recommendation Modeling For Clothing Product Selection Recommendation System* ini berfokus pada pengembangan model rekomendasi berbasis pengetahuan untuk sistem pemilihan produk pakaian di Simple Inc Store, sebuah toko di Kawasan pasar Klewer, Jawa Tengah. Tujuan dari penelitian ini adalah untuk membantu pelanggan dalam memilih produk pakaian dengan lebih efisien mengingat banyaknya pilihan yang tersedia. Metode yang digunakan adalah RAD (*Rapid Application Development*) dan pendekatan rekomendasi berbasis pengetahuan yang menghitung nilai kesamaan antara kebutuhan pelanggan dan atribut produk. Kekurangan dari penelitian ini adalah penelitian ini masih mengandalkan sistem rekomendasi yang bersifat konvensional, tanpa integrasi dengan teknologi modern seperti machine learning yang dapat meningkatkan akurasi rekomendasi. Selain itu, penelitian ini belum melakukan pengujian lebih lanjut untuk mengevaluasi efektivitas sistem rekomendasi dalam situasi dunia nyata dan umpan balik dari pengguna [28].

1.9.7 *The Effect of Electronic Human Resources Management on Organizational Health of Telecommunication companies in Jordan*

Penelitian yang berjudul *The Effect of Electronic Human Resources Management on Organizational Health of Telecommunication companies in Jordan* ini membahas mengenai pengaruh penggunaan *Electronic Human Resource*

Management (e-HRM) terhadap Kesehatan organisasi di perusahaan telekomunikasi di Yordania. e-HRM merupakan contoh penggunaan teknologi informasi untuk meningkatkan praktik manajemen sumber daya manusia. Penelitian ini menekankan pentingnya pelatihan secara elektronik. Adanya sistem ini membantu perusahaan dalam mengumpulkan umpan balik dari karyawan. Penelitian ini menunjukkan bahwa penggunaan e-HRM ini memberikan dampak positif dalam meningkatkan kesehatan organisasi yang dapat dilihat dari peningkatan efisiensi, efektivitas, dan kepuasan kerja karyawan. Kekurangan dari penelitian ini adalah analisis yang dilakukan masih kurang mendalam karena analisis yang dilakukan tidak menangkap semua dinamika dan variabel yang berinteraksi dalam konteks kesehatan organisasi [5].

1.9.8 *A Survey on Complex Knowledge Base Question Answering : Methods, Challenges, and Solutions*

Penelitian yang berjudul *A Survey on Complex Knowledge Base Question Answering : Methods, Challenges, and Solutions* ini membahas mengenai KBQA (*Knowledge Base Question Answering*) dalam konteks pertanyaan kompleks yang melibatkan pemrosesan bahasa alami dan basis pengetahuan yang terstruktur. Terdapat dua metode yang digunakan dalam penelitian ini untuk mengidentifikasi tantangan yang dihadapi, yaitu metode berbasis *parsing semantic* (SP-Based) dan berbasis pengambilan informasi (IR-Based). Pengembangan *user manual* berbasis *website* dapat memanfaatkan teknologi KBQA untuk memberikan jawaban yang relevan dan cepat terhadap pertanyaan pengguna. Sistem ini dapat menyediakan akses cepat ke informasi yang relevan dan dapat mengurangi waktu pencarian informasi yang dibutuhkan. Kekurangan dari penelitian ini adalah evaluasi yang dilakukan belum cukup komprehensif untuk menggambarkan kinerja dari metode yang digunakan. Metode evaluasi yang lebih mendalam diperlukan untuk memberikan gambaran tentang efektivitas metode yang digunakan [29].

1.9.9 *Agile Software Development : Methodologies and Trends*

Penelitian yang berjudul *Agile Software Development : Methodologies and Trends* ini membahas tentang pengembangan perangkat lunak *Agile* dan metodologi

terkait dengan melihat nilai-nilai dan prinsip-prinsip utama dalam pengembangan perangkat lunak yang fleksibel. Penelitian ini menunjukkan bahwa pendekatan *Agile* dilakukan untuk mengatasi keterbatasan metode pengembangan tradisional dengan memberikan fleksibilitas dalam menangani perubahan persyaratan, serta meningkatkan kolaborasi antara pengembang dan pelanggan. Kekurangan dari penelitian ini adalah analisis yang kurang mendalam untuk mengeksplorasi tantangan yang dihadapi dalam menerapkan prinsip *Agile* dalam pengembangan [2].

1.9.10 *My Chatbot Companion – a Study of Human-Chatbot Relationships*

Penelitian yang berjudul *My Chatbot Companion – a Study of Human-Chatbot Relationships* ini membahas tentang hubungan antara manusia dan *chatbot* sosial, khususnya bagaimana hubungan ini berkembang dan dampaknya terhadap kehidupan pengguna. Penelitian ini menggunakan teori penetrasi sosial yang digunakan untuk memahami proses pengembangan HCR (*Human-Chatbot Relationships*) melalui wawancara dengan pengguna Replika, sebuah *chatbot* sosial. Pengguna melaporkan adanya dampak positif terhadap kesejahteraan pengguna. Karakteristik *chatbot* dianggap *non-judgemental* dan mempengaruhi pengembangan hubungan. Kekurangan dari penelitian ini adalah penelitian ini hanya menggunakan data kualitatif yang dapat menjadi subjektif dan sulit untuk diukur secara kuantitatif [11].

1.9.11 *Harmonising Knowledge for Safer Materials Via the “Nanocommons” Knowledge Base*

Penelitian yang berjudul *Harmonising Knowledge for Safer Materials Via the “NanoCommons” Knowledge Base* ini membahas tentang NanoCommons *Knowledge Base* yang merupakan sebuah infrastruktur riset berbasis komunitas untuk menyatukan data dan pengetahuan terkait *nano safety* (keamanan material nano). NanoCommons *knowledge base* berfungsi sebagai pusat informasi atau *knowledge base* yang menyatukan data dari berbagai proyek dan eksperimen terkait material nano. *Knowledge base* ini tidak hanya menyimpan data, tetapi juga menyediakan *semantic* untuk memahami, mencari, dan menggunakan data dengan

cara yang terstruktur dan terstandarisasi. *Platform* ini menggunakan pendekatan FAIR (*Findable, Accessible, Interoperable, Reusable*) dalam manajemen data ilmiah. Sistem yang dikembangkan merupakan contoh dari implementasi *web-based semantic knowledge base* yang mengintegrasikan data eksperimen, model prediktif, dan alat analisis dalam satu sistem terpadu. Penelitian ini dilakukan dengan tujuan untuk menyediakan interoperabilitas semantik dan teknis antara berbagai sumber data dan alat nanoinformatika, serta memfasilitasi akses terbuka terhadap data dan alat analisis prediktif terkait nanomaterial. Kekurangan dari penelitian ini adalah skalabilitas yang terbatas dikarenakan cakupan proyek NanoCommons masih terbatas pada sebagian kecil dari seluruh proyek nanomaterial yang ada [30].

1.9.12 *Extracting a Knowledge Base of Mechanism from COVID-19 Papers*

Penelitian yang berjudul *Extracting a Knowledge Base of Mechanism from COVID-19 Papers* ini membahas tentang pengembangan *knowledge base* dari mekanisme ilmiah yang ditemukan dalam literatur COVID-19, seperti aktivitas virus, dampak ekonomi, penggunaan AI dalam diagnosis, dan lainnya. Peneliti mengembangkan skema relasi mekanisme yang luas dan seragam, melatih model untuk mengekstraksi relasi dari makalah ilmiah, dan menyusun *knowledge base* bernama COMB (*COVID-19 Open Mechanism Knowledge Base*). Penelitian ini berfokus pada ekstraksi dan penyusunan *knowledge base* yang bertujuan untuk mendukung pencarian ilmiah yang lebih terstruktur dan akurat dibandingkan pencarian konvensional. COMB diimplementasikan sebagai antarmuka pencarian berbasis *web* yang dapat menerima *input query* berupa entitas dan hubungan, memberikan hasil berupa relasi yang ditemukan dalam literatur ilmiah, dan mengindeks entitas menggunakan *model embedding* dan pencarian berbasis FAISS. Kekurangan dari penelitian ini adalah model tergantung pada literatur yang tersedia sehingga jika informasi tidak ada dalam teks atau tidak disebutkan secara eksplisit, maka tidak dapat diekstraksi [31].

1.9.13 *Global Knowledge Base for Municipal Solid Waste Management : Framework Development and Application in Waste Generation Prediction*

Penelitian yang berjudul *Global Knowledge Base for Municipal Solid Waste Management : Framework Development and Application in Waste Generation Prediction* ini mengembangkan kerangka kerja untuk membangun *knowledge base* global mengenai pengelolaan sampah padat perkotaan (*Municipal Solid Waste/MSW*). Penelitian ini berfokus pada pengumpulan data global tentang jumlah, komposisi, praktik pengelolaan sampah, pengembangan model statistic untuk memprediksi produksi sampah berdasarkan faktor sosial ekonomi, iklim, serta menyusun *framework* yang menghubungkan berbagai pengetahuan terkait MSW, seperti kebijakan, teknologi, dan konteks sosial ekonomi. Sistem ini membangun *knowledge base* sebagai repositori sistematis dari data-data terkait MSW global, alat bantu untuk menghubungkan data-data agar lebih mudah digunakan oleh peneliti dan pembuat kebijakan, serta dasar bagi pengambilan keputusan berbasis bukti (*evidence-based decision-making*). Kekurangan dari penelitian ini adalah data belum terintegrasi secara penuh dikarenakan belum ada penggabungan data dengan aspek perilaku pengguna, misalnya preferensi warga terhadap daur ulang, dan lain-lain [32].

1.9.14 *Prompting as Probing : Using Language Models for Knowledge Base Construction*

Penelitian yang berjudul *Prompting as Probing : Using Language Models for Knowledge Base Construction* ini memperkenalkan metode ProP (*Prompting as Probing*) yang menggunakan model bahasa besar seperti GPT-3 untuk melakukan *Knowledge Base Construction* (KBC), yaitu konstruksi basis pengetahuan dari teks. Peneliti menciptakan *prompt-prompt* khusus untuk mengeksplorasi pengetahuan tersimpan di model bahasa dan mengevaluasi kualitas jawaban yang dihasilkan, terutama dalam konteks kompetisi LM-KBC 2022. Penelitian ini bertujuan untuk mengekstrak atau menyusun basis pengetahuan dari model bahasa berupa triple : subjek-predikat-objek. Model digunakan untuk menjawab pertanyaan berbasis triple dan GPT-3 harus mengisi objek secara akurat, layaknya basis pengetahuan

semantik seperti Wikidata. Pendekatan ini berpotensi besar diterapkan dalam *knowledge base*, seperti sistem pencarian jawaban, portal FAQ (*Frequently Asked Question*), atau *chatbot* dengan pemahaman fakta. Kekurangan dari penelitian ini adalah ketergantungan pada *prompt* manual sehingga tidak selalu skalabel dan model seperti GPT-3 tidak bisa menjawab pertanyaan terkait fakta terbaru pasca 2020 [33].

1.9.15 DeepKE : A Deep Learning Based Knowledge Extraction Toolkit for Knowledge Base Population

Penelitian yang berjudul *DeepKE : A Deep Learning Based Knowledge Extraction Toolkit for Knowledge Base Population* ini memperkenalkan DeepKE, sebuah *open-source toolkit* berbasis *deep learning* untuk *knowledge extraction* dalam rangka mengisi dan melengkapi *knowledge base*. DeepKE dirancang untuk berbagai skenario kompleks seperti *low-resource*, *document-level*, dan multimodal (teks dan gambar). *Toolkit* ini mendukung beberapa tugas ekstraksi informasi utama, yaitu *Named Entity Recognition* (NER), *Relation Extraction* (RE), dan *Attribute Extraction* (AE). *Toolkit* ini menyediakan model bawaan, *framework* terpadu, dan antarmuka *online* yang mempermudah pengguna untuk melakukan ekstraksi informasi secara *real-time*. DeepKE berkontribusi pada *Knowledge Base Population* (KBP), yaitu proses mengekstrak pengetahuan dari data yang tidak terstruktur, misalnya dokumen teks, untuk memperluas dan memperbarui isi dari *knowledge base*. DeepKE dapat diterapkan dalam sistem *knowledge base* untuk memperkaya konten secara otomatis. Kekurangan dari penelitian ini adalah ketergantungan pada skema yang telah ditentukan sehingga ekstraksi hanya berjalan efektif jika entitas atau relasi sudah terdapat dalam skema pelatihan [34].

1.9.16 Introducing a Method for Intervals Correction on Multiple Likert Scales : A Case Study on an Urban Soundscape Data Collection Instrument

Penelitian yang berjudul *Introducing a Method for Intervals Correction on Multiple Likert Scales : A Case Study on an Urban Soundscape Data Collection Instrument* ini membahas tentang penggunaan Skala Likert dalam meneliti *soundscape* di lingkungan perkotaan. Penelitian ini menggunakan dataset dari 984

penilaian *soundscape* di 11 lokasi di London. Setelah dilakukan analisis, ditemukan bahwa terdapat pola umum dalam cara responden menilai atribut suara. Salah satunya adalah terdapat kecenderungan responden lebih menilai suatu *soundscape* sebagai “menyenangkan” dibandingkan “menggangu”. Hal tersebut menunjukkan adanya ketidakseimbangan dalam interpretasi skala. Penelitian ini menemukan bahwa penerapan nilai metrik yang telah dikoreksi dapat meningkatkan akurasi model prediksi dan menunjukkan peningkatan signifikan dalam kinerjanya. Kekurangan dari penelitian ini adalah metode yang diperkenalkan tidak dapat digunakan pada semua konteks, tetapi diperlukan penyesuaian apakah metode tepat untuk digunakan [35].

BAB III. METODE PENELITIAN

3.1 Waktu dan Tempat Penelitian

3.1.1 Waktu

Tabel 3. Waktu Penelitian

No	Kegiatan Penelitian	Nov	Des	Jan	Feb	Mar	Apr	Mei	Jun
1.	<i>Requirements</i>								
2.	<i>Design</i>								
3.	<i>Implementation</i>								
4.	<i>Integration</i>								
5.	<i>Testing</i>								
6.	<i>Deployment</i>								
7.	<i>Iteration</i>								
8.	Pelaporan								

Tabel diatas merupakan waktu pelaksanaan kegiatan penelitian mulai dari tahap *planning* hingga pelaporan yang dilakukan selama 8 bulan, mulai dari November 2024 hingga Juni 2025. Iterasi dilakukan berdasarkan umpan balik yang diberikan oleh pengguna akhir. Proses kembali ke tahap sebelumnya berdasarkan perubahan *requirement* menyesuaikan dengan umpan balik pengguna akhir.

3.1.2 Tempat

Penelitian ini dilaksanakan di Universitas Lampung yang berlokasi di Jalan Prof. Dr. Ir. Sumantri Brojonegoro No. 1, Gedong Meneng, Kecamatan Rajabasa, Kota Bandar Lampung, Lampung dengan objek BPJS Ketenagakerjaan yang berlokasi di Jalan Gatot Subroto No. 79, Karet Semanggi, Kecamatan Setiabudi, Kota Jakarta Selatan, Jakarta.

3.2 Alat dan Bahan Penelitian

3.2.1 Alat Penelitian

Penelitian yang dilaksanakan menggunakan beberapa alat untuk menunjang seluruh tahapan yang telah dilaksanakan. Adapun alat pada penelitian ini terbagi menjadi dua jenis, yaitu perangkat keras dan perangkat lunak.

Tabel di bawah ini merupakan rincian alat – alat yang digunakan selama penelitian ini :

Tabel 4. Perangkat Keras

Perangkat Keras	Fungsi
Laptop AMD Ryzen 3 3250U with Radeon Graphics	Membuat desain dan implementasi desain ke kode program
Mouse	Menggerakkan kursor
Keyboard	Memasukkan / mengetikkan data

Tabel 5. Perangkat Lunak

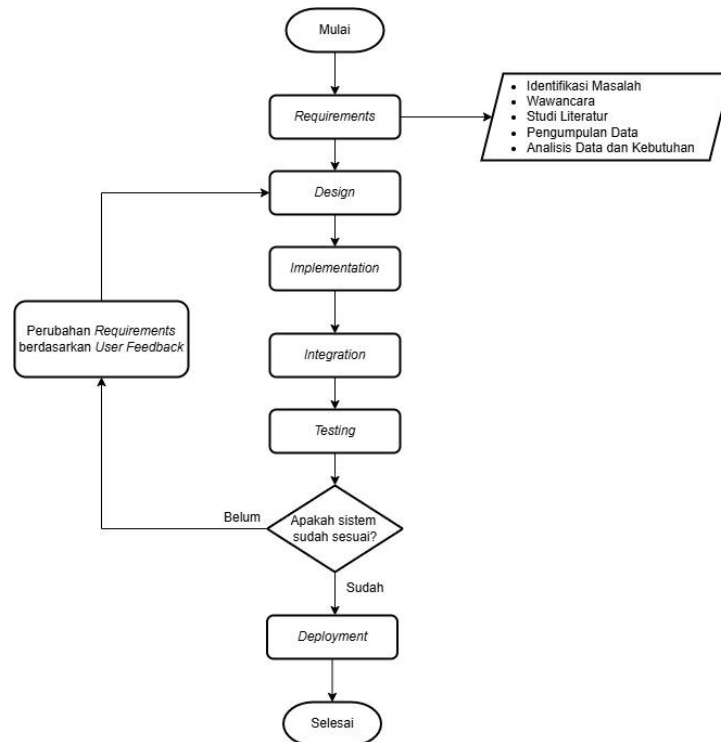
Perangkat Lunak	Fungsi
Visual Studio Code <i>version</i> 1.96.4	Melakukan pengembangan, <i>debugging</i> , dan kolaborasi untuk mengimplementasikan desain ke dalam kode program
MySQL <i>version</i> 10.6.22	Membantu dalam menambah, mengakses, dan memproses data yang disimpan dalam basis data.
phpMyAdmin <i>version</i> 5.2.2	Mengelola basis data MySQL.

XAMPP <i>version</i> 3.3.0	Membantu dalam pengujian aplikasi secara lokal.
Figma	Membuat desain tampilan aplikasi <i>web</i>
draw.io	Membuat diagram untuk desain aplikasi <i>web</i>
dbdiagram.io	Membuat rancangan database aplikasi <i>web</i>
<i>Framework</i> Bootstrap <i>version</i> 4.6	Membantu dalam pengembangan <i>frontend</i> aplikasi <i>web</i>
<i>Framework</i> Codeigniter <i>version</i> 4.6.0	Membantu dalam pengembangan <i>backend</i> aplikasi <i>web</i>
Diaglogflow Essentials	Membantu dalam pengembangan <i>chatbot</i>
Postman <i>version</i> 11.51.3	Membantu dalam <i>testing</i> bagian <i>backend</i>
API (<i>Application Programming Interface</i>)	Membantu dalam integrasi sistem, layanan, dan aplikasi yang digunakan
IDCloudHost	Membantu dalam <i>web hosting</i>

3.2.2 Bahan Penelitian

Penelitian yang dilaksanakan menggunakan data *user manual* aplikasi eksternal dan data *dummy* dari aplikasi internal BPJS Ketenagakerjaan. Penggunaan data *dummy* tersebut dikarenakan data yang digunakan merupakan data internal perusahaan yang memuat informasi bersifat privasi seperti data pengguna atau karyawan dan juga aplikasi internal yang hanya dapat diakses oleh karyawan internal perusahaan. Penelitian ini dilakukan dengan karyawan BPJS Ketenagakerjaan yang merupakan *end user* dari *website*. *Website* divalidasi oleh karyawan melalui kuesioner.

3.3 Tahapan Penelitian



Gambar 4. Tahapan Penelitian

3.3.1 Requirements

a) Identifikasi Masalah

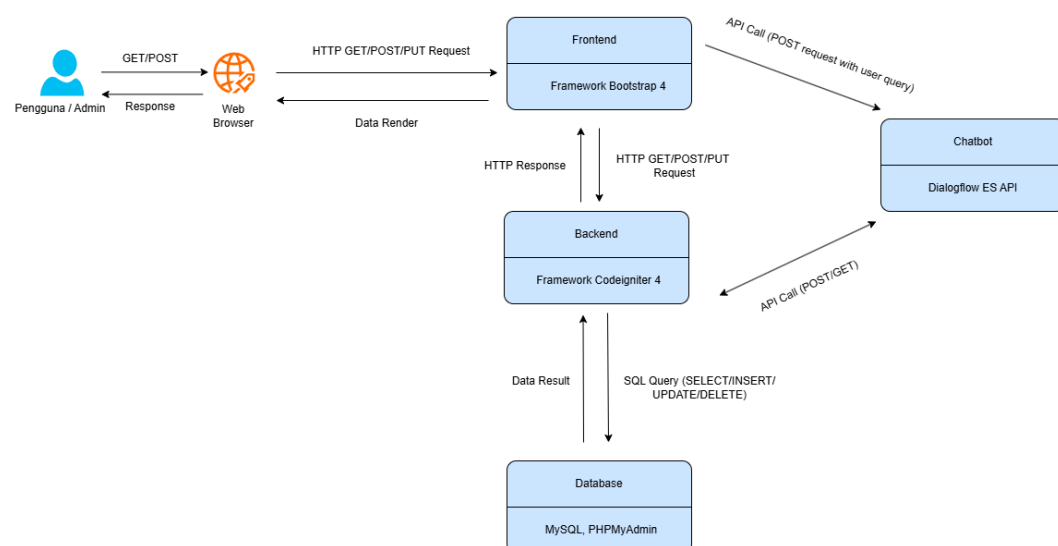
Pada tahap ini sudah dilakukan wawancara mengenai bagaimana sistem yang digunakan, pentingnya *knowledge base* berbasis *website*, dan keunggulannya dibandingkan dengan *user manual* dengan ekstensi *file* PDF atau Word untuk melakukan identifikasi kebutuhan pengguna, yaitu karyawan internal BPJS Ketenagakerjaan untuk menyesuaikan dengan sistem yang dibuat. Daftar pertanyaan pada metode wawancara yang telah dilakukan didapatkan dari studi literatur dengan membaca penelitian terkait. Proses ini dilakukan dengan melakukan diskusi dengan pemangku kepentingan untuk mendapatkan gambaran sistem yang dikembangkan. Berikut merupakan daftar pertanyaan dan jawabannya.

Tabel 6. Daftar Pertanyaan dan Jawaban Wawancara

No	Pertanyaan	Jawaban
1.	Bagaimana sistem dokumentasi <i>user manual</i> yang digunakan oleh BPJS Ketenagakerjaan saat ini?	Saat ini, dokumentasi <i>user manual</i> masih menggunakan <i>file</i> dengan ekstensi PDF atau Word.
2.	Apa kelemahan dari <i>penggunaan user manual</i> dalam ekstensi PDF atau Word?	Ekstensi tersebut dinilai kurang praktis, sulit diakses cepat, dan membutuhkan proses yang panjang saat ada pembaruan informasi.
3.	Mengapa sistem <i>knowledge base</i> berbasis <i>website</i> perlu dikembangkan?	<i>Knowledge base</i> berbasis <i>website</i> perlu dikembangkan karena sistem ini memudahkan karyawan menemukan informasi secara cepat, mudah diakses, dan lebih mudah untuk dilakukan pembaruan yang dapat dilakukan oleh admin.
4.	Apa keunggulan <i>knowledge base</i> berbasis <i>website</i> dibandingkan <i>user manual</i> PDF atau Word?	Keunggulannya adalah informasi lebih mudah dicari, dapat diperbarui secara langsung, dan tidak perlu mengirimkan ulang dokumen ke seluruh karyawan saat ada perubahan.
5.	Siapa pengguna utama dari sistem <i>knowledge base</i> ini?	Seluruh karyawan internal BPJS Ketenagakerjaan.
6.	Mengapa <i>knowledge base</i> dikembangkan dalam bentuk <i>website</i> ?	<i>Knowledge base</i> dikembangkan dalam bentuk <i>website</i> dikarenakan pekerjaan sehari-hari di BPJS Ketenagakerjaan berbasis <i>website</i>

		dan menggunakan PC kantor yang terhubung dengan jaringan internet internal.
--	--	---

Setelah menyelesaikan kegiatan MBKM, telah dilakukan diskusi kembali dengan mentor yang bertanggung jawab dengan pengembangan sistem ini. Hasil dari diskusi tersebut didapatkan bahwa perlunya pengembangan fitur baru, yaitu *chatbot* yang dapat membantu pengguna dalam berinteraksi dengan sistem layaknya berkomunikasi dengan manusia. Fitur ini dikembangkan menyesuaikan dengan permintaan *stakeholders* untuk memenuhi kebutuhan pengguna dalam mendapatkan informasi yang lebih mudah dan cepat. Berikut merupakan gambaran arsitektur sistem knowledge base berbasis *website* yang dikembangkan.



Gambar 5. Arsitektur Sistem

Gambar 5 merupakan arsitektur sistem dari *knowledge base* berbasis *website* yang dikembangkan. Pengguna atau admin berinteraksi dengan *frontend* yang dibangun menggunakan *framework* Bootstrap 4 melalui *web browser* dengan mengirimkan permintaan HTTP (GET, POST, PUT) untuk menampilkan data atau mengirimkan informasi. *Frontend* kemudian berkomunikasi dengan *backend*

berbasis *framework* Codeigniter 4 menggunakan permintaan HTTP (GET, POST, PUT) untuk memproses logika bisnis dan data. *Backend* selanjutnya berinteraksi dengan *database* (MySQL) menggunakan SQL *queries* (SELECT, INSERT, UPDATE, DELETE) untuk mengambil atau memanipulasi data, dan mengembalikan hasil ke *frontend*. Untuk fungsionalitas *chatbot*, *frontend* langsung memanggil *chatbot* (Dialogflow ES API) menggunakan API Call (POST request) yang berisi pertanyaan pengguna. Jika *chatbot* memerlukan data tambahan dari sistem (misalnya dari *knowledge base* yang disimpan di *database*), *chatbot* akan memicu API Call (POST/GET) kembali ke *backend* yang kemudian mengambil data dari *database* dan mengembalikannya ke *chatbot* untuk menghasilkan respons akhir yang dikirim kembali ke *frontend*. Setelah itu, *frontend* akan me-render data ke *web browser* dan menampilkannya ke pengguna atau admin.

b) Analisis Kebutuhan Sistem

Pengembangan *knowledge base* berbasis *website* diharapkan dapat memudahkan pengguna dalam mencari informasi mengenai penggunaan aplikasi yang ada di BPJS Ketenagakerjaan, baik aplikasi internal maupun eksternal. Pengembangan sistem ini memungkinkan admin yang merupakan pengelola dari *website* ini untuk dapat mengelola langsung *website* dengan meng-edit, menambahkan, dan menghapus *user manual* sehingga dapat memberikan pengguna informasi yang lebih terbaru dan akurat.

a) Kebutuhan Fungsional

Tabel 7. Kebutuhan Fungsional

ID	Kebutuhan Fungsional	Deskripsi
KF-01	Antarmuka Pengguna	Antarmuka dapat diakses dengan baik di berbagai perangkat. (<i>desktop dan mobile</i>) dan pengguna dapat dengan mudah menemukan

		informasi yang diperlukan melalui fitur yang tersedia.
KF-02	Pencarian dan Filter	Pengguna dapat mencari informasi dengan menggunakan fitur <i>search</i> dan <i>chatbot</i> , serta dapat menggunakan fitur <i>filter</i> untuk menampilkan daftar <i>user manual</i> aplikasi internal atau eksternal.
KF-03	Informasi Berbasis Artikel	Admin dapat menambahkan, meng- <i>edit</i> , dan menghapus artikel dengan mudah, serta format informasi yang tersedia variatif.

b) Kebutuhan Non-Fungsional

Tabel 8. Kebutuhan Non-Fungsional

ID	Kebutuhan Non-Fungsional	Deskripsi
KNF-01	Kinerja	Sistem harus memberikan waktu respons yang cepat untuk memenuhi permintaan pengguna dan dapat diakses kapan saja.
KNF-02	Keamanan	Pengguna harus melalui proses otentikasi untuk mengakses <i>website</i> dan menjamin bahwa data pengguna dilindungi melalui enkripsi.

KNF-03	Kegunaan	Desain harus intuitif dan mudah digunakan sehingga pengguna dapat dengan cepat menemukan informasi tanpa kesulitan.
--------	----------	---

3.3.2 *Design*

Pada tahap ini dilakukan pembuatan desain untuk *website* yang dikembangkan. Tahap ini dimulai dengan menentukan *userflow* dari *website* yang dikembangkan. *Website* dapat diakses oleh karyawan BPJS Ketenagakerjaan dan admin yang mengelola *website* tersebut. Perbedaan dari kedua pengguna tersebut adalah karyawan hanya dapat melihat isi *website*, sedangkan admin dapat melakukan CRUD (*Create, Read, Update, Delete*) pada *website*.

A. UML (Unified Modelling Language)

a. Use Case Diagram



Gambar 6. Use Case Diagram

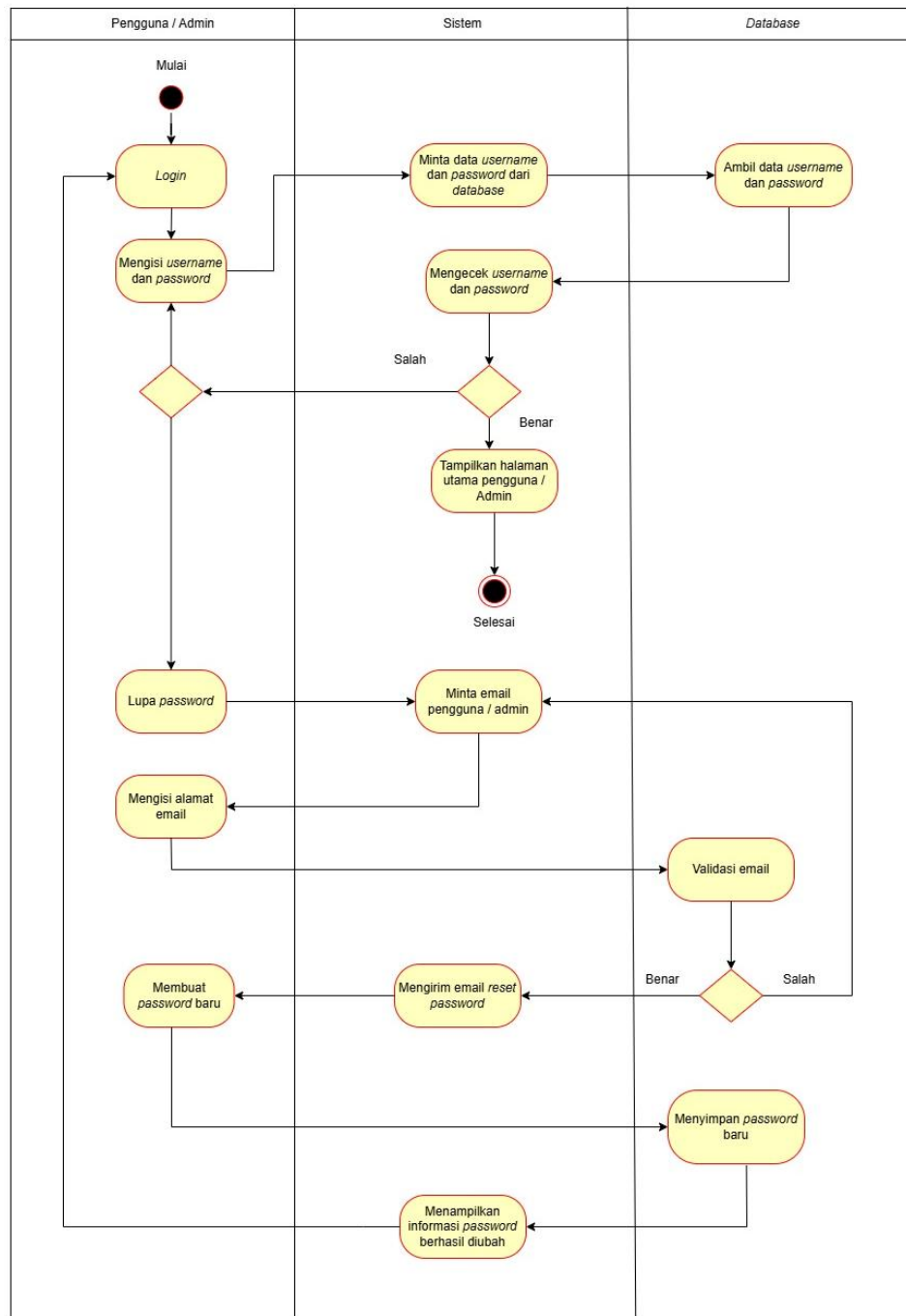
Gambar 6 merupakan *use case diagram* dari sistem yang dikembangkan. Sistem ini terdiri dari dua aktor yang akan dijelaskan pada tabel berikut.

Tabel 9. Aktor dalam *Knowledge Base* berbasis *Website* dan Fungsinya

Aktor	Fungsi
Pengguna dan Admin	Melakukan <i>login</i> , mencari informasi, membaca <i>user manual</i> , dan <i>logout</i>
Pengguna	Menggunakan <i>chatbot</i> dan mem- <i>filter user manual</i>
Admin	Mengelola <i>user manual</i> dengan menambah, meng- <i>edit</i> , dan menghapus <i>user manual</i> , serta melihat <i>history</i>

b. Activity Diagram

a) Login

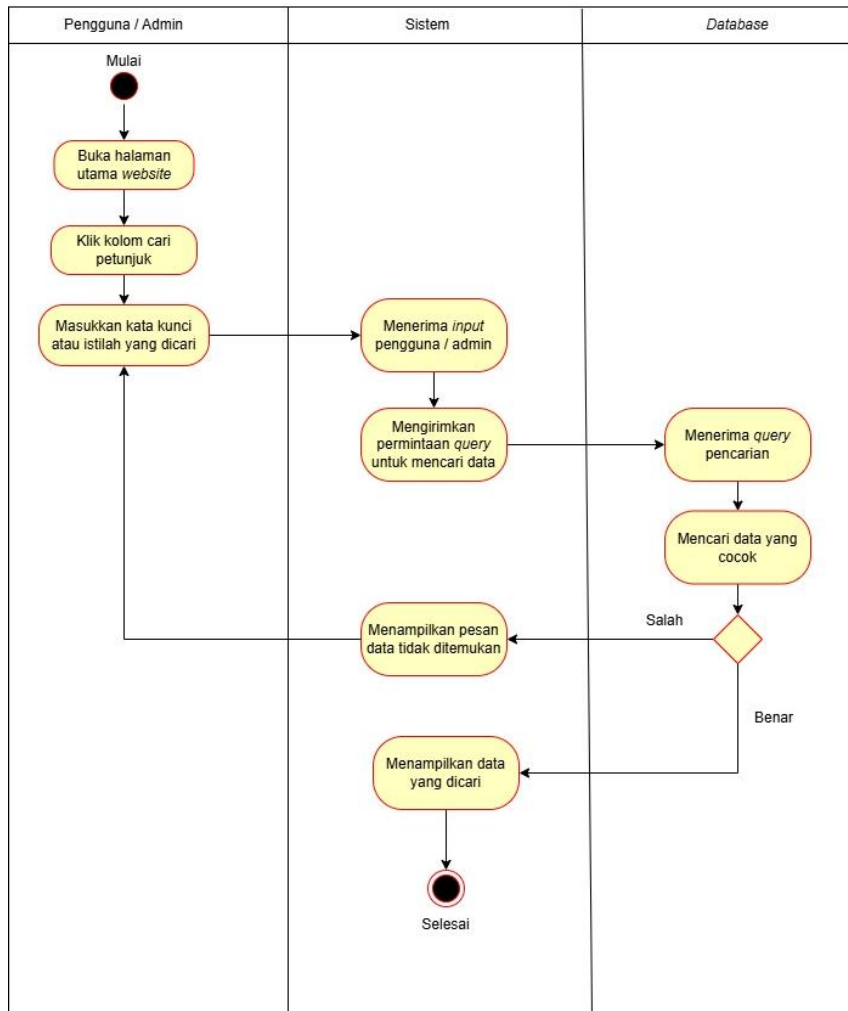


Gambar 7. Login

Gambar 7 merupakan *activity diagram* yang menjelaskan mengenai aktivitas *login* pengguna atau admin. Aktivitas dimulai dengan pengguna atau admin mengisi *username* dan *password*. Sistem akan minta data *username* dan

password dari *database* dan sistem *database* akan mengambil data *username* dan *password*. Kemudian, sistem akan mengecek *username* dan *password*. Jika *username* dan *password* benar, maka sistem akan menampilkan *landing page* pengguna atau admin dan aktivitas selesai. Jika salah, maka pengguna atau admin harus mengisi ulang *username* dan *password*. Jika pengguna atau admin klik lupa *password*, maka sistem akan meminta *email* pengguna atau admin dan pengguna atau admin dapat mengisi alamat *email* untuk mengatur ulang *password*. Setelah itu sistem pada *database* akan memvalidasi alamat *email* yang telah diisi apakah sudah terdaftar atau belum. Jika sudah terdaftar, maka sistem akan mengirim *email reset password* dan pengguna atau admin dapat membuat *password* baru. Setelah itu, *password* baru akan disimpan dalam *database* dan sistem akan menampilkan informasi *password* berhasil diubah. Kemudian, pengguna atau admin dapat melakukan *login* kembali untuk masuk ke dalam *website*.

b) Mencari Informasi

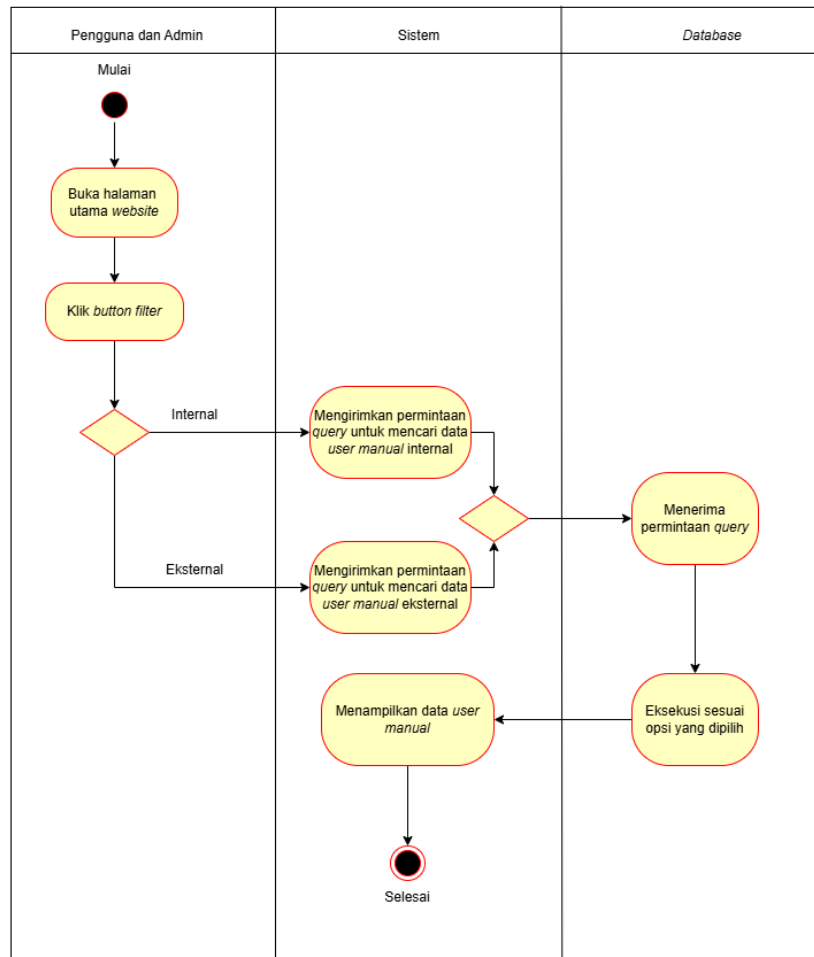


Gambar 8. Mencari Informasi

Gambar 8 merupakan *activity diagram* yang menjelaskan aktivitas mencari informasi yang dapat dilakukan oleh pengguna / admin. Aktivitas dimulai dengan membuka halaman utama *website* dan kolom cari petunjuk di bagian atas halaman. Kemudian, pengguna atau admin dapat memasukkan kata kunci atau istilah yang ingin dicari. Sistem akan menerima *input* dari pengguna atau admin dan mengirimkan permintaan *query* ke sistem *database* untuk mencari data. Kemudian, sistem pada *database* akan menerima *query* pencarian dan mencari data yang cocok. Jika terdapat data yang cocok, maka sistem akan menampilkan data yang dicari dan aktivitas selesai. Jika tidak terdapat data yang cocok, maka sistem akan

menampilkan pesan data tidak ditemukan dan pengguna atau admin dapat memasukkan kata kunci atau istilah yang dicari kembali.

c) Mem-filter user manual

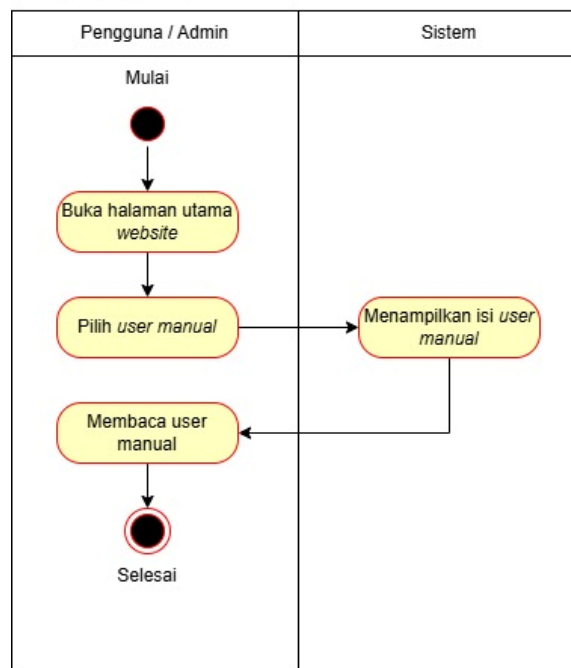


Gambar 9. Mem-filter User Manual

Gambar 9 merupakan *activity diagram* dari fitur *mem-filter user manual* yang dapat dilakukan oleh pengguna dan admin. Aktivitas dimulai dengan membuka halaman utama *website* dan pengguna serta admin dapat klik *button filter*. Jika pengguna dan admin klik *button internal* saja, maka sistem akan mengirimkan permintaan *query* untuk mencari data *user manual* yang masuk dalam kategori internal saja. Jika pengguna dan admin klik *button eksternal* saja, maka sistem akan mengirimkan permintaan *query* untuk mencari data *user manual* yang masuk dalam

kategori eksternal saja. Setelah itu, sistem pada *database* akan menerima permintaan *query* dan mengeksekusi sesuai dengan opsi yang dipilih. Kemudian, sistem akan menampilkan data *user manual* dan aktivitas selesai. Jika pengguna dan admin tidak klik *button* internal atau eksternal maupun klik kedua *button*, maka sistem akan menampilkan semua data *user manual*.

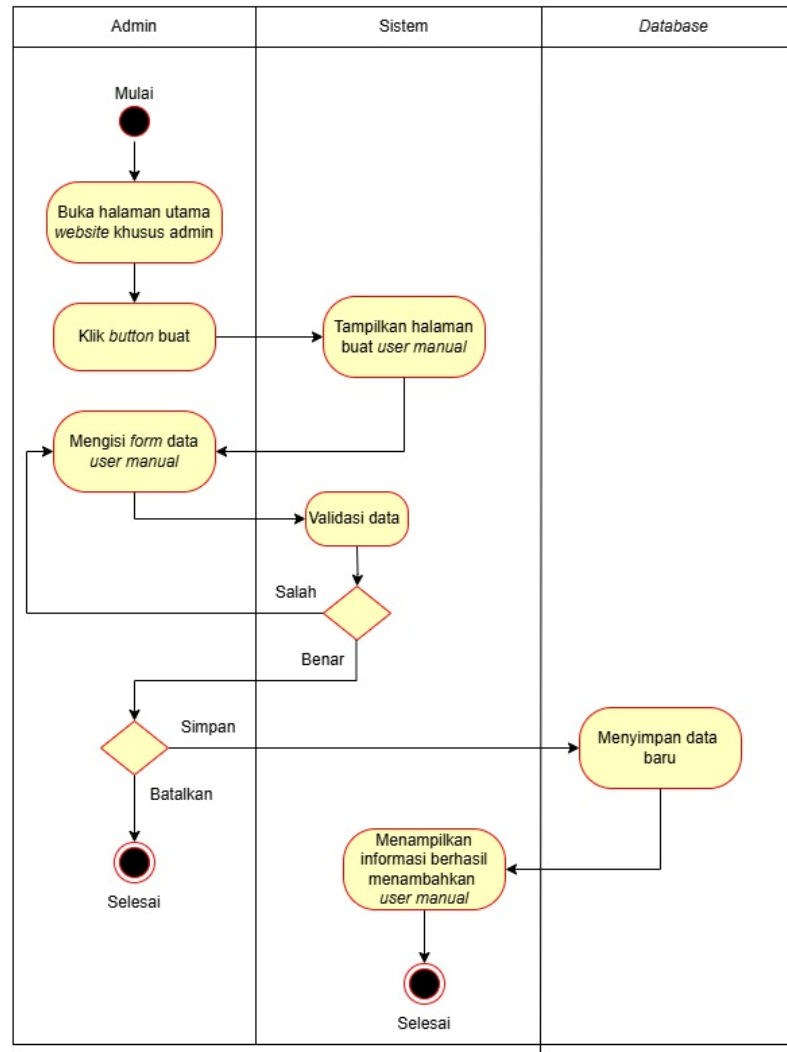
d) **Membaca User Manual**



Gambar 10. Membaca *User Manual*

Gambar 10 merupakan *activity diagram* dari membaca *user manual*. Aktivitas dimulai dengan membuka halaman utama *website*. Setelah itu pengguna atau admin dapat memilih *user manual* yang ingin diakses. Sistem akan menampilkan isi dari *user manual* yang telah dipilih dan pengguna dapat membaca isi dari *user manual* untuk mendapat informasi yang diinginkan dan aktivitas selesai.

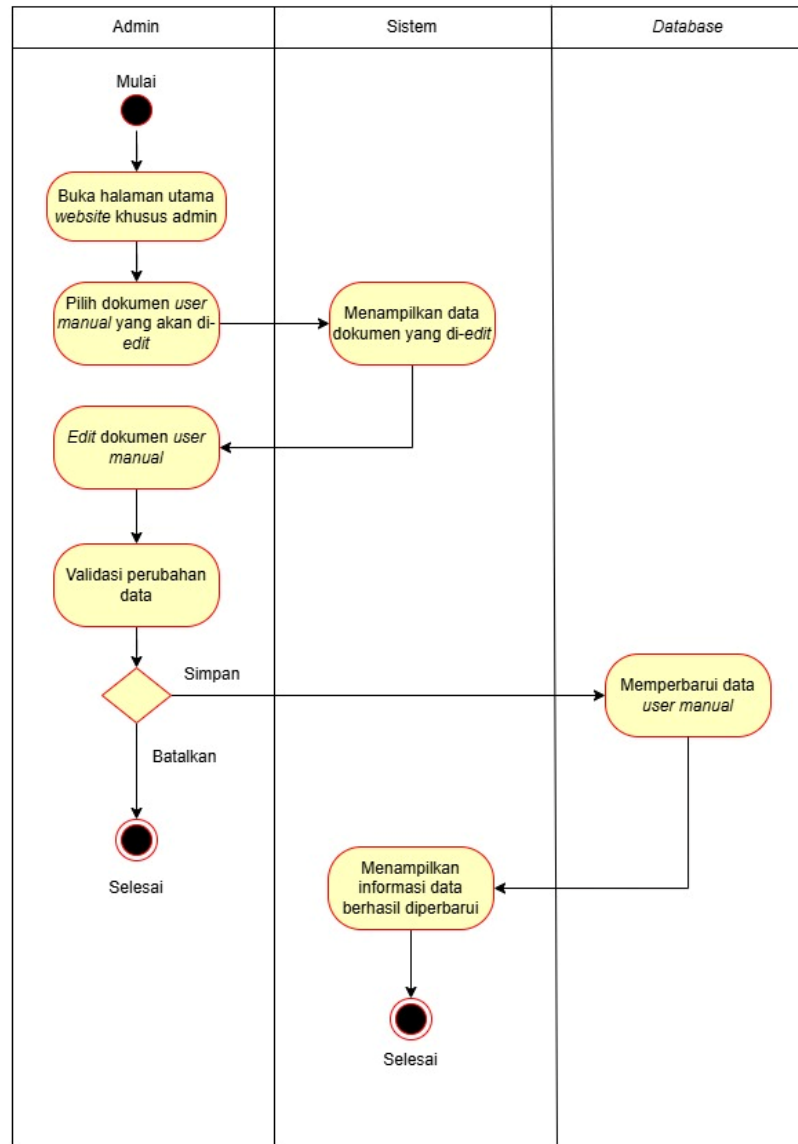
e) **Mengelola *User Manual* : Menambahkan *User Manual***



Gambar 11. Mengelola *User Manual*: Menambahkan *User Manual*

Gambar 11 merupakan gambar aktivitas mengelola *user manual* yang hanya dapat dilakukan oleh admin, yaitu menambahkan *user manual*. Aktivitas dimulai dengan membuka halaman utama *website* khusus admin dan klik *button* buat. Lalu, sistem akan menampilkan halaman buat *user manual*. Admin dapat mengisi *form* data *user manual* dan sistem akan memvalidasi data yang sudah diisi. Jika benar, maka data yang sudah diisi akan disimpan dalam *database* dan sistem akan menampilkan informasi berhasil menambahkan *user manual* dan aktivitas selesai. Jika salah, maka admin akan kembali ke halaman *form* data *user manual* dan dapat mengisi datanya kembali.

f) **Mengelola User Manual : Meng-edit User Manual**

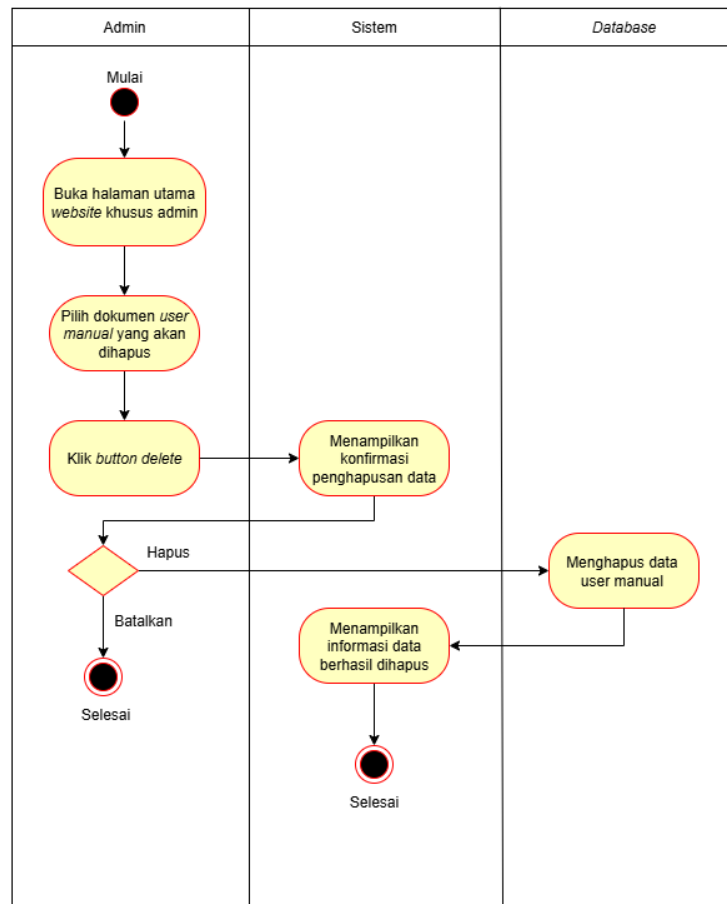


Gambar 12. Mengelola User Manual: Meng-edit User Manual

Gambar 12 merupakan *activity diagram* dari aktivitas mengelola *user manual* yang hanya dapat dilakukan oleh admin, yaitu *meng-edit user manual*. Aktivitas dimulai dengan membuka halaman utama *website* khusus admin. Admin dapat memilih dokumen *user manual* yang akan di-*edit*. Kemudian, sistem akan menampilkan data dokumen yang di-*edit* dan admin dapat meng-*edit* dokumen *user manual* tersebut. Setelah itu, sistem akan memvalidasi perubahan data. Admin dapat klik simpan untuk menyimpan data yang sudah di-*edit* dan *database* akan diperbarui. Sistem akan menampilkan informasi data berhasil diperbarui dan

aktivitas selesai. Admin juga dapat klik batalkan untuk membatalkan perubahan dan aktivitas selesai.

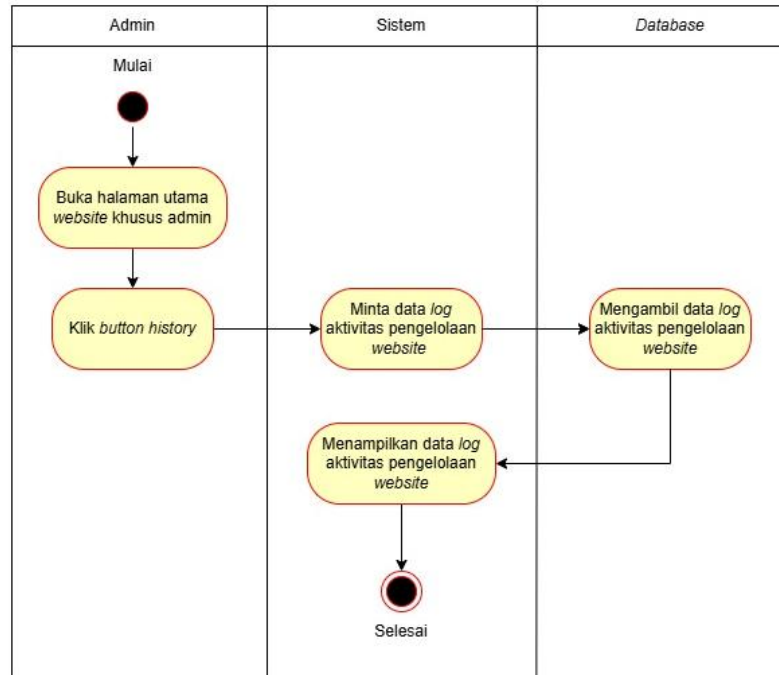
g) Mengelola User Manual : Menghapus User Manual



Gambar 13. Mengelola User Manual : Menghapus User Manual

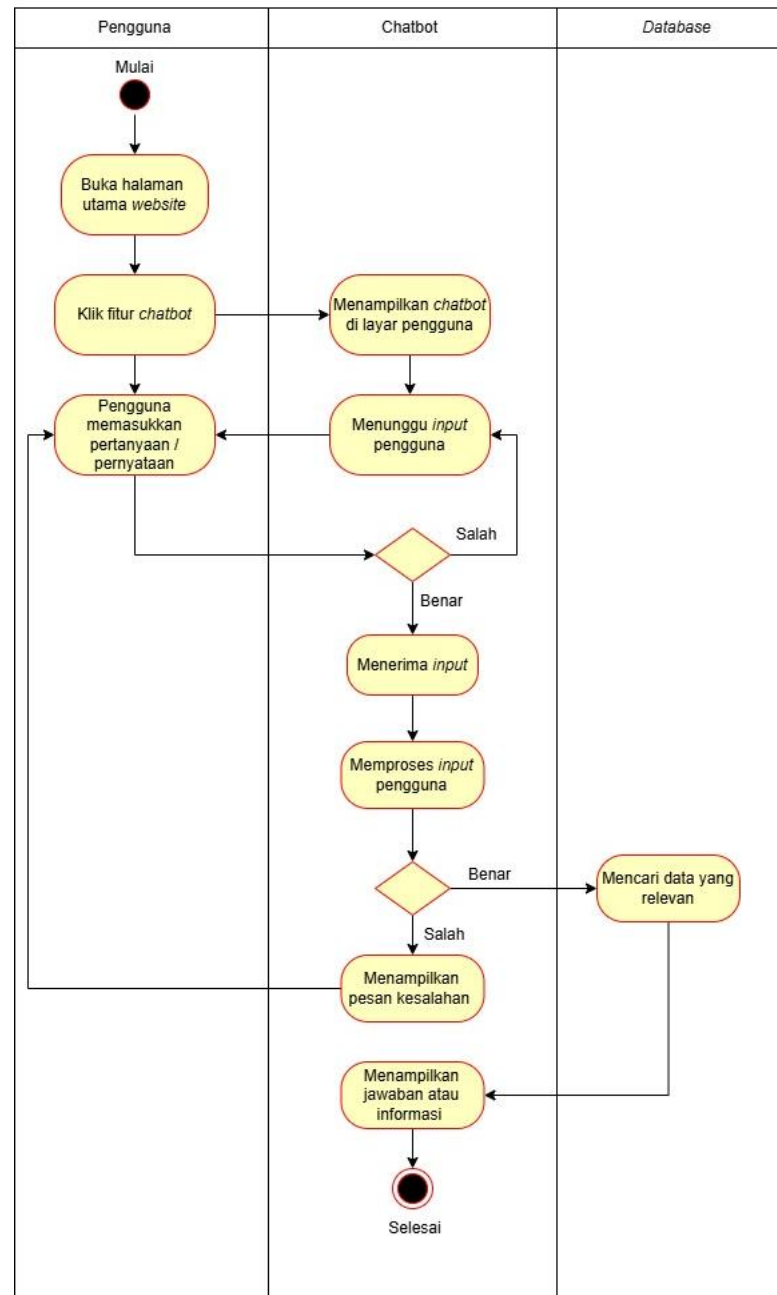
Gambar 13 merupakan *activity diagram* dari aktivitas mengelola *user manual* yang hanya dapat dilakukan oleh admin, yaitu menghapus *user manual*. Aktivitas dimulai dengan membuka halaman utama *website* khusus admin. Admin dapat memilih dokumen *user manual* yang akan dihapus dan klik *button delete*. Kemudian, sistem akan menampilkan konfirmasi penghapusan data. Admin dapat klik hapus dan data *user manual* akan dihapus dalam *database*. Sistem akan menampilkan informasi data berhasil dihapus dan aktivitas selesai. Admin juga dapat klik batalkan untuk membatalkan penghapusan *user manual* dan aktivitas selesai.

h) Mengelola *User Manual* : Melihat *History*



Gambar 14. Mengelola *User Manual* : Melihat *History*

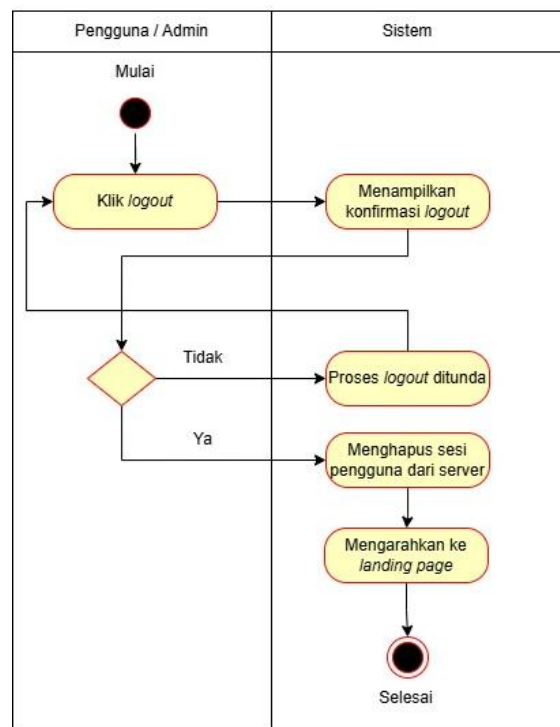
Gambar 14 merupakan *activity diagram* dari aktivitas mengelola *user manual* yang hanya dapat dilakukan oleh admin, yaitu melihat *history*. Aktivitas dimulai dengan membuka halaman utama *website* khusus admin dan admin dapat klik *button history* untuk melihat *history* pengelolaan *website*. Kemudian, sistem akan meminta data *log* aktivitas pengelolaan *website* dan sistem pada *database* akan mengambil data *log* aktivitas tersebut. Setelah itu, sistem akan menampilkan data *log* aktivitas pengelolaan *website* yang telah dilakukan dan aktivitas selesai.

i) *Chatbot*Gambar 15. *Chatbot*

Gambar 15 merupakan *activity diagram* dari fitur *chatbot* yang dapat digunakan oleh pengguna untuk menemukan informasi. Aktivitas dimulai dengan pengguna membuka halaman utama *website* dan klik fitur *chatbot*. Sistem *chatbot* akan menampilkan *chatbot* di layar pengguna dan akan menunggu *input* dari pengguna. Jika *input* benar, maka *input* akan diterima dan diproses. Jika *input* salah,

maka *chatbot* akan menunggu *input* pengguna kembali dan pengguna diminta untuk memasukkan pertanyaan atau pernyataan kembali. Setelah *input* diproses, jika *input* benar atau dikenali, maka *chatbot* akan berinteraksi dengan *database* untuk mencari data yang relevan. Jika *input* salah atau tidak dikenali, maka *chatbot* akan menampilkan pesan kesalahan dan pengguna dapat memasukkan pertanyaan atau pernyataan kembali. Setelah mendapatkan data yang relevan, *chatbot* akan menampilkan jawaban atau informasi yang dibutuhkan oleh pengguna dan aktivitas selesai.

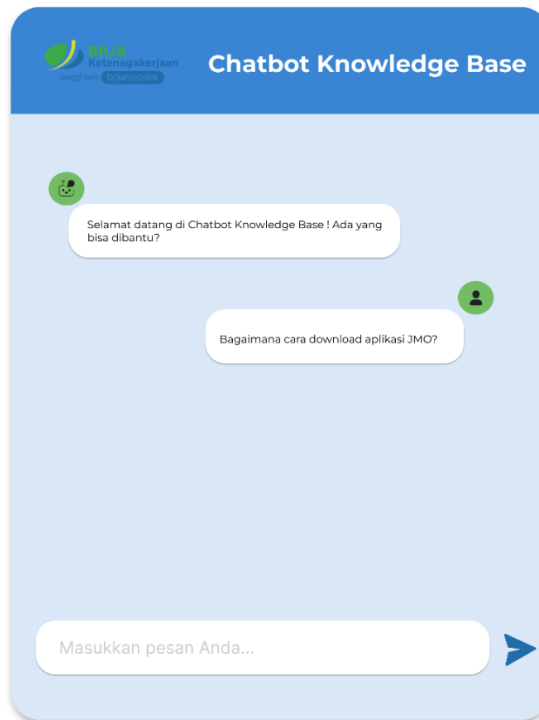
j) Logout



Gambar 16. Logout

Gambar 16 merupakan *activity diagram* dari *logout*. Aktivitas dimulai dengan pengguna atau admin klik *logout*. Kemudian, sistem akan menampilkan konfirmasi *logout*. Jika pengguna atau admin memilih ya, maka sistem akan menghapus sesi pengguna dari *server* dan sistem akan mengarahkan pengguna atau admin ke *landing page* dan aktivitas selesai. Jika pengguna atau admin memilih tidak, maka proses *logout* akan ditunda.

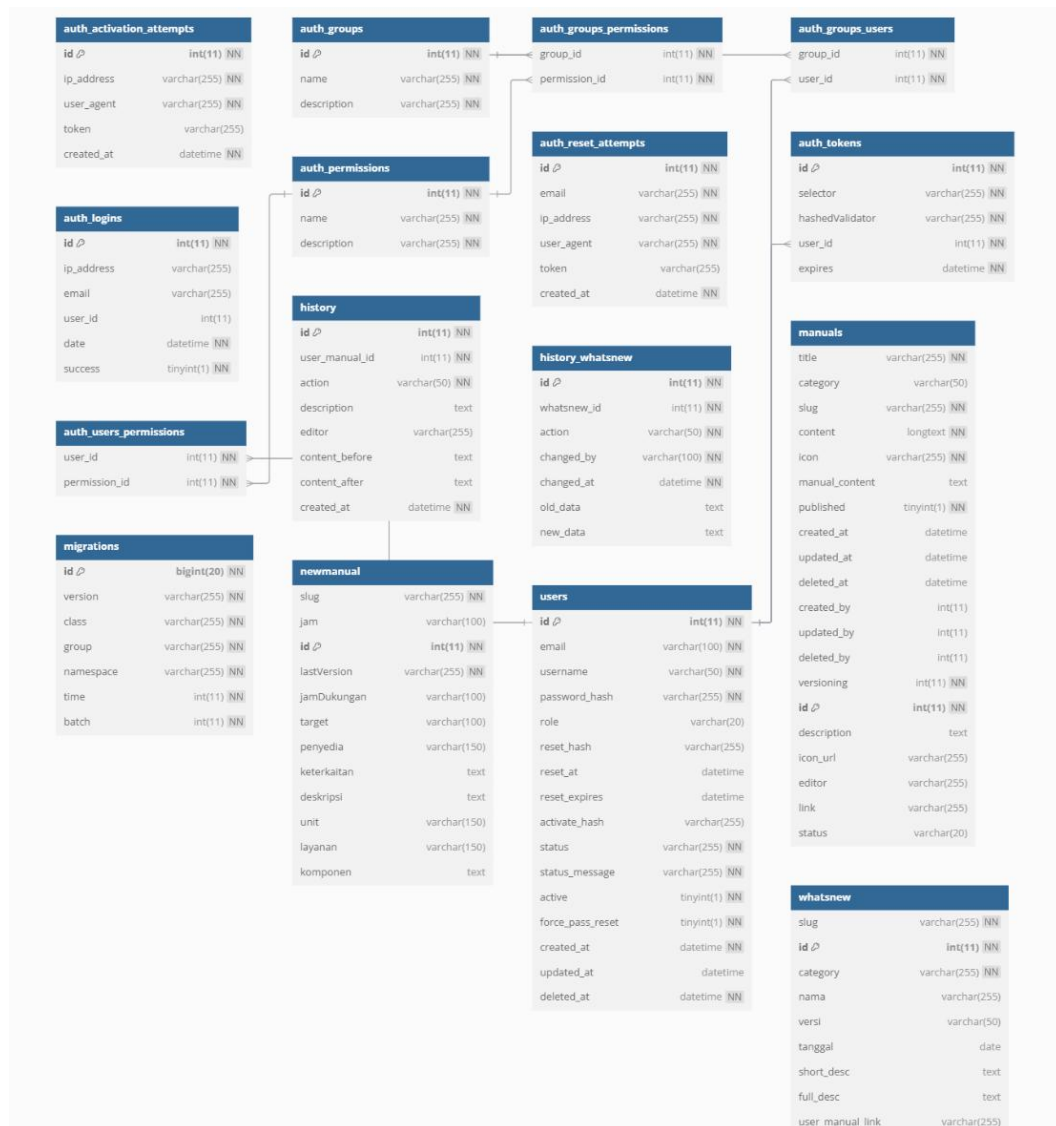
B. Desain Tampilan Fitur Baru



Gambar 17. Desain Tampilan *Chatbot*

Gambar 17 merupakan desain dari fitur baru yang akan dikembangkan, yaitu fitur *chatbot*. Fitur ini berfungsi untuk membantu pengguna mendapatkan informasi dengan lebih mudah dan cepat dengan memasukkan pertanyaan atau pernyataan. Sistem akan menjawab pertanyaan atau pernyataan dari pengguna sesuai dengan data yang ada pada *website*. Fitur *chatbot* dapat membantu pengguna berinteraksi dengan sistem layaknya berkomunikasi dengan manusia sehingga pengguna dapat menemukan informasi yang diinginkan. Pengembangan *chatbot* menggunakan aplikasi Dialogflow Essential. Proses dilakukan dengan melakukan instalasi aplikasi, membuat desain pertanyaan dan jawaban yang digunakan dalam *chatbot*, dan mengintegrasikannya. Integrasi dilakukan menggunakan API sehingga *chatbot* yang dibuat di Dialogflow Essentials dapat digunakan dalam *website knowledge base*.

C. Rancangan Database



Gambar 18. Rancangan Database

Gambar 18 merupakan rancangan *database* dari sistem yang akan dikembangkan. Rancangan *database* di atas dibuat menggunakan bantuan dbdiagram.io untuk merancang basis data dari sistem. Diagram ini terdiri dari beberapa kelompok tabel utama, yaitu tabel autentikasi, tabel manajemen *user manual*, tabel *history* perubahan, dan tabel metadata tambahan. Selain tabel utama untuk manajemen pengguna, hak akses, data manual, pada rancangan *database* ini terdiri dari tabel autentikasi pendukung untuk mencatat aktivitas keamanan akun pengguna, yaitu *auth_activation_attempts*, *auth_logins*, *auth_reset_attempts*, dan

auth_tokens. Tabel *migrations* digunakan untuk menyimpan *history* migrasi skema *database* yang dilakukan. Tabel-tabel tersebut saling terhubung melalui *relasi one-to-many* dan *many-to-many* menyesuaikan dengan fungsionalitas sistem. Tabel *users* memiliki relasi dengan beberapa tabel, yaitu tabel *auth_logins* berelasi ke *users* melalui kolom *user_id* untuk mencatat riwayat *login* setiap pengguna. Tabel *auth_tokens* menyimpan *user_id* untuk menghubungkan token autentikasi dengan pengguna. Tabel *auth_reset_attempts* dan *auth_activation_attempts* tidak memiliki relasi langsung ke tabel *users* namun tetap menggunakan *email* sebagai identitas pengguna yang sedang melakukan percobaan aktivasi atau *reset password*. Sistem manajemen otorisasi menggunakan relasi *many-to-many* yang dimediasi oleh tabel pivot. Tabel *auth_groups* dan *auth_permissions* dihubungkan melalui tabel *auth_group_permissions* sehingga sebuah grup bisa memiliki banyak hak akses dan sebaliknya. Tabel *users* dengan *auth_groups* dihubungkan melalui tabel *auth_groups_users* yang memungkinkan seorang pengguna tergabung dalam satu atau lebih grup. Selain itu, tabel *users* juga berelasi dengan *auth_permissions* melalui tabel *auth_users_permissions* sehingga *user* bisa memiliki hak akses tertentu secara langsung di luar grup. Tabel *manuals* menyertakan atribut *created_by* dan *updated_by* yang merujuk ke *users.id* untuk mencatat siapa yang membuat dan memperbaiki data. Tabel menyertakan *user_manual_id* merujuk kepada ke *manuals.id*, dan *editor* merujuk ke *users.id* untuk identifikasi pengguna yang melakukan perubahan. Tabel *history_whatsnew* berelasi ke tabel *whatsnew* melalui kolom *whatsnew_id* untuk mencatat *history* perubahan informasi fitur baru.

Table	Action	Rows	Type	Collation	Size	Overhead
auth_activation_attempts	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	16.0 KIB	-
auth_groups	Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_general_ci	16.0 KIB	-
auth_groups_permissions	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	48.0 KIB	-
auth_groups_users	Browse Structure Search Insert Empty Drop	5	InnoDB	utf8mb4_general_ci	48.0 KIB	-
auth_logins	Browse Structure Search Insert Empty Drop	110	InnoDB	utf8mb4_general_ci	48.0 KIB	-
auth_permissions	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	16.0 KIB	-
auth_reset_attempts	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	16.0 KIB	-
auth_tokens	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	48.0 KIB	-
auth_users_permissions	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	48.0 KIB	-
manuals	Browse Structure Search Insert Empty Drop	4	InnoDB	utf8mb4_general_ci	48.0 KIB	-
migrations	Browse Structure Search Insert Empty Drop	11	InnoDB	utf8mb4_general_ci	16.0 KIB	-
newmanual	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	16.0 KIB	-
users	Browse Structure Search Insert Empty Drop	5	InnoDB	utf8mb4_general_ci	16.0 KIB	-
whatsnew	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	16.0 KIB	-
14 tables	Sum	137	InnoDB	utf8mb4_general_ci	416.0 KIB	0 B

Gambar 19. Database

Gambar 19 merupakan struktur *database* dari sistem yang dikembangkan. Terdapat beberapa tabel yang terdiri dari tabel dengan prefix *auth_*, yaitu *library* autentikasi Codeigniter, serta tabel *manuals*, *migrations*, *newmanual*, *users*, dan *whatsnew*. Tabel-tabel dengan prefix *auth_* yang terdiri dari *auth_activation_attempts* untuk mencatat usaha aktivasi akun, *auth_groups* untuk menyimpan daftar grup pengguna, *auth_groups_permissions* untuk mencatat hubungan antara grup dan hak akses, *auth_group_users* untuk mencatat relasi antara user dan grupnya, *auth_logins* untuk *log* atau catatan *login* pengguna, *auth_permissions* untuk mencatat daftar hak akses yang bisa diatur, *auth_tokens* untuk menyimpan token autentikasi, dan *auth_users_permissions* untuk mencatat hak akses spesifik ke pengguna. Tabel *manuals* digunakan untuk menyimpan data terkait *user manual* aplikasi yang terdaftar dalam sistem. Tabel *migrations* digunakan untuk menyimpan informasi tentang perubahan skema *database* secara otomatis oleh sistem migrasi bawaan Codeigniter. Tabel *newmanual* digunakan untuk menyimpan informasi deskripsi aplikasi. Tabel *users* digunakan untuk menyimpan informasi dasar pengguna. Tabel *whatsnew* digunakan untuk fitur informasi *update* terbaru atau pengumuman sistem, seperti *changelog* atau fitur baru.

3.3.3 *Implementation*

Pada tahap ini dilakukan implementasi desain yang telah disetujui menjadi kode yang dapat dijalankan. *Frontend* dikembangkan dengan menggunakan *framework* Bootstrap 4. *Backend* dikembangkan menggunakan *framework* Codeigniter 4. Fitur terbaru, yaitu *chatbot* dikembangkan menggunakan Dialogflow Essentials. Proses dilakukan dengan mengembangkan fitur dan fungsionalitas sesuai dengan rencana yang telah ditetapkan.

3.3.4 *Integration*

Pada tahap ini dilakukan proses penggabungan berbagai komponen dan sistem menjadi satu kesatuan yang dapat berfungsi sesuai dengan kebutuhan pengguna. Proses dilakukan dengan menggabungkan kode yang telah ditulis sehingga dapat berfungsi dengan baik. Integrasi sistem juga dilakukan dengan mengintegrasikan *frontend* dengan *backend* serta *chatbot* menggunakan API sehingga sistem dapat diakses dan dikelola dengan baik.

3.3.5 *Testing*

Pada tahap ini dilakukan pengujian untuk memeriksa sistem yang telah dikembangkan. Proses ini dilakukan dengan menggunakan menggunakan kuesioner dan metode UAT (*User Acceptance Testing*) untuk menguji apakah sistem yang dibuat sudah berfungsi dengan baik dan sesuai dengan kebutuhan pengguna.

3.3.6 *Deployment*

Pada tahap ini dilakukan transfer kode dan data ke lingkungan produksi dan menyiapkan *server* dan infrastruktur yang diperlukan untuk menjalankan *website* sehingga *website* dapat diakses oleh pengguna.

3.3.7 *Iteration*

Pada tahap ini dilakukan pengulangan siklus pengembangan untuk meningkatkan sistem berdasarkan umpan balik yang telah diterima. Umpan balik yang telah diterima digunakan untuk memperbarui *backlog* produk dengan menambahkan atau mengubah sistem sesuai dengan kebutuhan pengguna.

BAB V. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Adapun kesimpulan berdasarkan hasil penelitian yang telah dilakukan adalah sebagai berikut :

1. Pengembangan *Knowledge Base* berbasis *Website* telah berhasil dilakukan menggunakan *framework* Codeigniter 4 dan Bootstrap 4 dimana pengguna *website*, yaitu karyawan BPJS Ketenagakerjaan, dapat mencari dan mendapatkan informasi mengenai penggunaan aplikasi serta admin dapat melakukan pengelolaan *website* dengan baik.
2. Berdasarkan hasil pengujian menggunakan *User Acceptance Testing* (UAT) yang telah dilakukan oleh 2 *role* pengguna *website*, yaitu admin dan pengguna didapatkan skor aktual, yaitu 82 % sehingga dapat disimpulkan bahwa pengembangan *knowledge base* berbasis *website* sudah sesuai dengan kebutuhan pengguna dimana iterasi telah berhasil dilakukan sebanyak tiga kali untuk perbaikan berdasarkan umpan balik dari pengguna, yaitu perubahan pada desain, perbaikan pada *chatbot*, dan halaman *user manual* dikarenakan sebelumnya terdapat *bug* dan kesalahan konfigurasi sehingga belum memenuhi kebutuhan pengguna. Hasil akhir dari iterasi yang dilakukan berupa umpan balik dari pengguna bahwa sistem telah memenuhi kebutuhan pengguna.

5.2 Saran

Adapun saran berdasarkan penelitian yang telah dilakukan adalah sebagai berikut:

1. Pengembangan untuk aplikasi *mobile* dapat dilakukan untuk pengembangan selanjutnya agar aplikasi dapat diakses dengan lebih mudah melalui perangkat *mobile* tanpa harus mengakses *web browser*.
2. Menambahkan fitur notifikasi untuk pengguna jika terdapat pembaruan informasi agar pengguna mengetahui jika terdapat informasi terbaru.
3. Mengembangkan fitur pencarian dengan menambahkan proses normalisasi input agar sistem dapat menangani variasi kata kunci dengan lebih baik dan meningkatkan akurasi hasil pencarian.
4. Berdasarkan hasil pengembangan fitur *chatbot* pada *website knowledge base* menggunakan aplikasi Dialogflow Essentials, terdapat kekurangan pada aplikasi yang menyebabkan keterbatasan dimana respons yang diberikan oleh sistem belum sepenuhnya sesuai dengan harapan pengguna. Oleh karena itu, penggunaan Dialogflow Essentials untuk pengembangan *chatbot* perlu ditinjau kembali.

DAFTAR PUSTAKA

- [1] K. C. Laudon and J. P. Laudon, "Management information systems: managing the digital firm," *Rev. adm. contemp.*, vol. 7, no. 1, pp. 223–223, 2014, doi: 10.1590/S1415-65552003000100014.
- [2] S. Alsaqqa, S. Sawalha, and H. Abdel-Nabi, "Agile Software Development: Methodologies and Trends," *Int. J. Interact. Mob. Technol.*, vol. 14, no. 11, p. 246, Jul. 2020, doi: 10.3991/ijim.v14i11.13269.
- [3] N. A. Y. Arimbawa and S. W. M. Edi, "Implementasi Knowledge Management pada Perancangan Website Panduan Perusahaan menggunakan Framework Laravel," *jimik*, vol. 5, no. 3, pp. 2634–2646, Sep. 2024, doi: 10.35870/jimik.v5i3.945.
- [4] A. Singh, K. Ramasubramanian, and S. Shivam, *Building an Enterprise Chatbot: Work with Protected Enterprise Data Using Open Source Frameworks*. Berkeley, CA: Apress, 2019. doi: 10.1007/978-1-4842-5034-1.
- [5] A. AlHamad *et al.*, "The effect of electronic human resources management on organizational health of telecommuni-cations companies in Jordan," *10.5267/j.ijdns*, vol. 6, no. 2, pp. 429–438, 2022, doi: 10.5267/j.ijdns.2021.12.011.
- [6] M. Deryck, "Knowledge Base Systems in Practice: Approaches, Application Areas and Limitations," 2022.
- [7] D. C. Kung, *Software Engineering, Second Edition*. United States of America: McGraw Hill LLC, 2024.
- [8] R. F. Rose, *Software Development Activity Cycles: Collaborative Development, Continuous Testing and User Acceptance*. Berkeley, CA: Apress, 2022. doi: 10.1007/978-1-4842-8239-7.
- [9] J. S. Wrench, C. Thomas-Maddox, V. P. Richmond, and J. C. McCroskey, *Quantitative research methods for communication: a hands-on approach*, Third edition. New York Oxford: Oxford University Press, 2016.
- [10] R. Juan Ferdinand Hutaauruk, "Analisis User Experience Website Penggerak Jaminan Sosial Indonesia (PERISAI) BPJS Ketenagakerjaan Purwokerto

- menggunakan Metode User Acceptance Test (UAT),” *jati*, vol. 8, no. 3, pp. 3478–3483, May 2024, doi: 10.36040/jati.v8i3.9529.
- [11] M. Skjuve, A. Følstad, K. I. Fostervold, and P. B. Brandtzaeg, “My Chatbot Companion - a Study of Human-Chatbot Relationships,” *International Journal of Human-Computer Studies*, vol. 149, p. 102601, May 2021, doi: 10.1016/j.ijhcs.2021.102601.
- [12] Google Cloud, “*Dialogflow ES Documentation*,” [Online]. Available: <https://cloud.google.com/dialogflow/es/docs?hl=id> [Accessed : Mar 10, 2025]
- [13] A. Del Sole, “Introducing Visual Studio Code,” in *Visual Studio Code Distilled*, Berkeley, CA: Apress, 2019, pp. 1–17. doi: 10.1007/978-1-4842-4224-7_1.
- [14] B. Christudas, “MySQL,” in *Practical Microservices Architectural Patterns*, Berkeley, CA: Apress, 2019, pp. 877–884. doi: 10.1007/978-1-4842-4501-9_27.
- [15] The phpMyAdmin Project, “*phpMyAdmin Documentation*,” [Online]. Available: <https://www.phpmyadmin.net/> [Accessed : Jan 10, 2025].
- [16] S. F. Ramadhan and U. Rusmawan, *Membangun Aplikasi dengan PHP, Codeigniter, dan AJAX*. Jakarta: PT Elex Media Komputindo, 2018.
- [17] diagram.net, “*draw.io - Diagrams for Everyone*,” [Online]. Available: <https://www.diagrams.net/> [Accessed : Jan 10, 2025].
- [18] dbdiagram.io, “*Database Relationship Diagrams Design Tool*,” [Online]. Available: <https://dbdiagram.io/> [Accessed : Jan 10, 2025].
- [19] F. Staiano, *Designing and Prototyping Interfaces with Figma*. UK: Packt Publishing Ltd., 2022.
- [20] M. Lambert, *Learning Bootstrap 4*, Second. UK: Packt Publishing Ltd., 2016.
- [21] IDCloudHost, “*IDCloudHost - Cloud Hosting Indonesia*,” [Online]. Available: <https://idcloudhost.com/> [Accessed : May 20, 2025].
- [22] Isnardi, Ikhsan, and R. Asmara, *Membangun RestFull API Menggunakan Codeigniter 4 dan Client Android dengan Bahasan Pemrograman Kotlin*. Padang: Pustaka Galeri Mandiri, 2021.

- [23] D. Westerveld, *API testing and development with Postman: a practical guide to creating, testing, and managing APIs for automated software testing*. Birmingham Mumbai: Packt Publishing, 2021.
- [24] R. T. Ariadi, I. Kusumaning P, M. Hani'ah, M. Astiningrum, and V. Nur W, "Pembuatan User Manual Berbasis Java sebagai Tutorial Aplikasi Sistem Informasi pada Kader Posyandu Rajawali Singosari." Oktober 2021. [Online]. Available: <https://dspace.uui.ac.id/handle/123456789/33933>
- [25] I. Nurkasanah, Q. S. Alami, M. Er, R. Indraswari, R. P. Wibowo, and R. Lakoro, "Platform Knowledge-Based Website untuk Meningkatkan Visibilitas UMKM di Sektor Perikanan," *SWGT*, vol. 7, no. 3, Mar. 2023, doi: 10.12962/j26139960.v7i3.482.
- [26] N. Muzayyana, A. B. Salsabil, and N. N. Ayni, "Perancangan Sistem Rekomendasi Perbelanjaan item Kursi Kayu Pada Toko Online Berbasis Website dengan Metode Knowledge Based Recommendation," 2023.
- [27] C. Lastiko and B. S. Wicaksono, "Rancang Bangun Aplikasi Knowledge Management pada Pelayanan Jasa Berbasis Web dengan Metode Agile Development (Studi Kasus : PT. Cakrawala Indonesia Sejahtera)," vol. 1, no. 4, 2023.
- [28] V. Atina and D. Hartanti, "Knowledge Based Recommendation Modeling for Clothing Product Selection Recommendation System," *J. Tek. Inform. (JUTIF)*, vol. 3, no. 5, pp. 1407–1413, Oct. 2022, doi: 10.20884/1.jutif.2022.3.5.584.
- [29] Y. Lan, G. He, J. Jiang, J. Jiang, W. X. Zhao, and J.-R. Wen, "A Survey on Complex Knowledge Base Question Answering: Methods, Challenges and Solutions," May 25, 2021, *arXiv*: arXiv:2105.11644. doi: 10.48550/arXiv.2105.11644.
- [30] D. Maier *et al.*, "Harmonising knowledge for safer materials via the 'NanoCommons' Knowledge Base," *Front. Phys.*, vol. 11, p. 1271842, Nov. 2023, doi: 10.3389/fphy.2023.1271842.
- [31] T. Hope *et al.*, "Extracting a Knowledge Base of Mechanisms from COVID-19 Papers," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language*

- Technologies*, Online: Association for Computational Linguistics, 2021, pp. 4489–4503. doi: 10.18653/v1/2021.naacl-main.355.
- [32] R. He *et al.*, “Global knowledge base for municipal solid waste management: Framework development and application in waste generation prediction,” *Journal of Cleaner Production*, vol. 377, p. 134501, Dec. 2022, doi: 10.1016/j.jclepro.2022.134501.
- [33] D. Alivanistos, S. B. Santamaría, M. Cochez, J.-C. Kalo, E. van Krieken, and T. Thanapalasingam, “Prompting as Probing: Using Language Models for Knowledge Base Construction,” Jun. 19, 2023, *arXiv*: arXiv:2208.11057. doi: 10.48550/arXiv.2208.11057.
- [34] N. Zhang *et al.*, “DeepKE: A Deep Learning Based Knowledge Extraction Toolkit for Knowledge Base Population,” Sep. 18, 2023, *arXiv*: arXiv:2201.03335. doi: 10.48550/arXiv.2201.03335.
- [35] M. Lionello, F. Aletta, A. Mitchell, and J. Kang, “Introducing a Method for Intervals Correction on Multiple Likert Scales: A Case Study on an Urban Soundscape Data Collection Instrument,” *Front. Psychol.*, vol. 11, p. 602831, Jan. 2021, doi: 10.3389/fpsyg.2020.602831.