

**PENGEMBANGAN *FRONTEND WEBSITE* KLASIFIKASI SPESIES LEBAH
MADU INDO-MALAYA MENGGUNAKAN METODE *RAPID APPLICATION
DEVELOPMENT* (RAD)**

(Skripsi)

**Oleh
ARSWENDO ERZA SADEWA
2215061062**



**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS LAMPUNG
2026**

**PENGEMBANGAN *FRONTEND WEBSITE* KLASIFIKASI SPESIES
LEBAH MADU INDO-MALAYA MENGGUNAKAN METODE *RAPID
APPLICATION DEVELOPMENT* (RAD)**

Oleh

ARSWENDO ERZA SADEWA

Skripsi

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

Pada

**Jurusan Teknik Elektro
Fakultas Teknik Universitas Lampung**



**FAKULTAS TEKNIK
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG**

2026

ABSTRAK

PENGEMBANGAN *FRONTEND WEBSITE* KLASIFIKASI SPESIES LEBAH MADU INDO-MALAYA MENGGUNAKAN METODE *RAPID APPLICATION DEVELOPMENT* (RAD)

Oleh

ARSWENDO ERZA SADEWA

Identifikasi spesies lebah madu secara tradisional memiliki keterbatasan dalam hal kecepatan dan kebutuhan akan keahlian khusus dalam mengenali ciri fisik lebah secara mendalam. Meskipun model *Convolutional Neural Network* (CNN) telah terbukti akurat dalam klasifikasi citra, efektivitasnya dalam praktik lapangan terbatas tanpa adanya dukungan antarmuka (*frontend*) yang mudah diakses oleh pengguna. Penelitian ini bertujuan untuk merancang dan membangun *frontend website* klasifikasi spesies lebah madu Indo-Malaya menggunakan metode *Rapid Application Development* (RAD) yang diintegrasikan dengan prinsip *User Centered Design* (UCD). Sistem dikembangkan dengan menggunakan *framework* Next.js dan Shadcn UI, serta terintegrasi dengan model CNN di sisi *backend* melalui RESTful API. Hasil pengujian fungsional dengan *Black Box Testing* menunjukkan seluruh fitur sistem berjalan dengan valid. Pengujian performa menggunakan Google Lighthouse menghasilkan skor pada rentang 93 hingga 99 yang mengindikasikan optimasi pemuatan halaman sudah berjalan dengan baik. Selain itu, pengujian kegunaan dengan metode *System Usability Scale* (SUS) terhadap 15 responden yang terdiri dari peternak lebah dan masyarakat umum menghasilkan skor rata-rata 83,83, yang menempatkan sistem pada kategori *Acceptable* dengan *grade* B dan predikat *Excellent*. Penerapan metode RAD terbukti efektif dalam mempercepat siklus pengembangan sistem sekaligus menghasilkan platform klasifikasi dan informasi yang responsif dan efisien. Sistem ini berhasil memberikan kemudahan bagi peternak lebah maupun masyarakat umum dalam melakukan identifikasi secara instan dan akurat, serta mendapatkan informasi mengenai lebah madu secara terpusat dalam satu platform digital yang dapat diakses melalui beevra.web.id.

Kata Kunci: *Frontend*, Lebah Madu, *Rapid Application Development* (RAD), *User Centered Design* (UCD), Next.js, *Usability*.

ABSTRACT

FRONTEND WEBSITE DEVELOPMENT FOR INDO-MALAYA HONEY BEE SPECIES CLASSIFICATION USING THE RAPID APPLICATION DEVELOPMENT (RAD) METHOD

By

ARSWENDO ERZA SADEWA

Traditional identification of honey bee species has limitations in terms of speed and the requirement for specialized expertise in recognizing physical characteristics in depth. Although Convolutional Neural Network (CNN) models have proven accurate in image classification, their effectiveness in field practice is limited without the support of a user-friendly interface (frontend). This research aims to design and build a frontend website for Indo-Malayan honey bee species classification using the Rapid Application Development (RAD) method integrated with User-Centered Design (UCD) principles. The system was developed using the Next.js framework and Shadcn UI, and is integrated with a CNN model on the backend via a RESTful API. Functional testing results using Black Box Testing showed that all system features operate validly. Performance testing using Google Lighthouse produced scores in the range of 93 to 99, indicating that page loading optimization is performing well. Furthermore, usability testing using the System Usability Scale (SUS) method with 15 respondents, consisting of beekeepers and the general public, resulted in an average score of 83.83, placing the system in the “Acceptable” category with a “Grade B” and an “Excellent” rating. The implementation of the RAD method proved effective in accelerating the system development cycle while producing a responsive and efficient classification and information platform. This system successfully provides convenience for beekeepers and the general public in performing instant and accurate identification, as well as obtaining centralized information about honey bees within a single digital platform accessible via beevra.web.id.

Keywords: Frontend, Honey Bee, Rapid Application Development (RAD), User Centered Design (UCD), Next.js, Usability.

Judul Skripsi : PENGEMBANGAN *FRONTEND WEBSITE*
KLASIFIKASI SPESIES LEBAH MADU INDO-
MALAYA MENGGUNAKAN METODE *RAPID*
APPLICATION DEVELOPMENT (RAD)

Nama Mahasiswa : **Arswendo Erza Sadewa**

Nomor Pokok Mahasiswa : 2215061062

Program Studi : Teknik Informatika

Fakultas : Teknik

MENYETUJUI

1. Komisi Pembimbing

Pembimbing Utama

Pembimbing Pendamping

Dr. Eng. Ir. Mardiana, S.T., M.T., I.P.M. NIP 197203161999032002

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

2. Mengetahui

Ketua Jurusan Teknik Elektro

Ketua Program Studi Teknik
Informatika

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

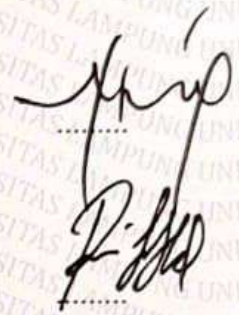
Yessi Mulyani, S.T., M.T.
NIP 197312262000122001

MENGESAHKAN

1. Tim Penguji

Ketua

: Dr. Eng. Ir. Mardiana, S.T., M.T., I.P.M.



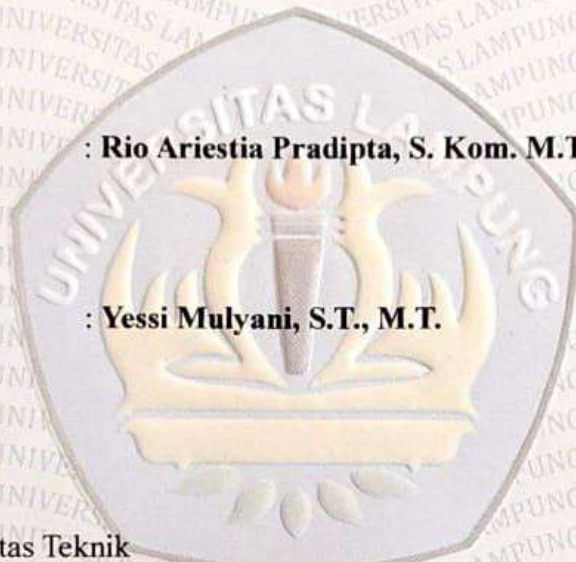
Sekretaris

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



Penguji

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



2. Dekan Fakultas Teknik



Dr. Ahmed Herison, S.T., M.T.
NIP 19691030200001001

Tanggal Lulus Ujian Skripsi : 04 Juni 2026

SURAT PERNYATAAN

Dengan ini saya menyatakan bahwa skripsi berjudul “Pengembangan *Frontend Website* Klasifikasi Spesies Lebah Madu Indo-Malaya Menggunakan Metode *Rapid Application Development* (RAD)” sepenuhnya merupakan hasil karya saya sendiri. Apabila di kemudian hari terbukti pernyataan ini tidak benar, saya siap menerima sanksi sesuai dengan ketentuan hukum yang berlaku.



Bandar Lampung, 04 Juni 2026


Arswendo Erza Sadewa
NPM. 2215061062

RIWAYAT HIDUP



Penulis terlahir sebagai putra ketiga dari pasangan Bapak Tukirno dan Ibu Sutarmi di Desa Banjar Agung Mataram, Lampung Tengah, pada tanggal 9 Oktober 2004. Pendidikan dasar diselesaikan di SD Negeri 3 Banjar Agung pada tahun 2016, kemudian menempuh jenjang sekolah menengah pertama di SMP Negeri 1 Bandar Mataram hingga lulus tahun 2019, dan menamatkan masa sekolah menengah atas di SMA Xaverius Pringsewu tahun 2022. Pada tahun yang sama, penulis resmi tercatat sebagai mahasiswa Program Studi Teknik Informatika, Jurusan Teknik Elektro, Universitas Lampung melalui jalur Seleksi Bersama Masuk Perguruan Tinggi Negeri (SBMPTN). Selama menempuh masa perkuliahan, penulis aktif mengembangkan kompetensi dan berorganisasi melalui beberapa kegiatan, di antaranya:

1. Menjadi peserta program Studi Independen Bersertifikat yang diselenggarakan oleh Kementerian Pendidikan dan Kebudayaan pada mitra PT Lentera Bangsa Benderang (Binar Academy) di tahun 2024.
2. Berperan sebagai Asisten Laboratorium Teknik Digital sejak tahun 2023 yang bertugas memberikan bimbingan teknis serta mendampingi mahasiswa dalam pelaksanaan pada 3 kegiatan mata kuliah praktikum.
3. Berkontribusi aktif dalam Himpunan Mahasiswa Teknik Elektro (HIMATRO) dengan mengemban amanah sebagai Kepala Divisi Sosial pada tahun 2024.
4. Menjalankan fungsi sebagai Staf Ahli Bidang Advokasi dan Kesejahteraan Mahasiswa pada organisasi Badan Eksekutif Mahasiswa (BEM) Fakultas Teknik Universitas Lampung untuk periode tahun 2025.

MOTTO

Segala perkara dapat kutanggung di dalam Dia yang memberi kekuatan kepadaku.

(Filipi 4:13)

Hidup bukan untuk saling mendahului. Untukmu, cintai diri sendiri hari ini.

Temukan makna hidupmu sendiri.

(Hindia)

*Kupersembahkan karya ini kepada Ayahanda dan
Ibunda terkasih, serta seluruh keluarga besar tercinta.*

UCAPAN TERIMA KASIH

Puji syukur ke hadirat Tuhan Yang Maha Esa, karena berkat dan kasih karunia-Nya, sehingga skripsi dengan judul “Pengembangan *Frontend Website* Klasifikasi Spesies Lebah Madu Indo-Malaya Menggunakan Metode *Rapid Application Development* (RAD)” ini dapat diselesaikan. Karya ini dipersembahkan sebagai wujud syukur dan dedikasi atas selesainya masa studi untuk meraih gelar Sarjana Teknik pada Jurusan Teknik Elektro, Fakultas Teknik, Universitas Lampung.

Keberhasilan dalam menuntaskan tugas akhir ini bukanlah hasil kerja keras mandiri, melainkan sebuah harmoni dari doa, bimbingan serta dukungan yang tak terputus dari berbagai pihak. Rasa hormat dan terima kasih yang sedalam-dalamnya disampaikan kepada:

1. Ibu Prof. Dr. Ir. Lusmeilia Afriani, D.E.A., I.P.M., selaku Rektor Universitas Lampung.
2. Bapak Dr. Ahmad Herison, S.T., M.T., selaku Dekan Fakultas Teknik Universitas Lampung.
3. Ibu Herlinawati, S.T., M.T., selaku Ketua Jurusan Teknik Elektro Fakultas Teknik Universitas Lampung.
4. Ibu Yessi Mulyani, S.T., M.T., selaku Ketua Program Studi Teknik Informatika sekaligus Dosen Penguji, atas ketegasan, kritik dan masukan berharganya yang telah menempa kualitas karya ini hingga mencapai titik terbaiknya.
5. Bapak Puput Budi Wintoro, S.Kom., M.T.I., selaku Dosen Pembimbing Akademik yang telah menjadi kompas serta pemberi arahan, motivasi dan bimbingannya selama masa perkuliahan di Teknik Informatika Universitas Lampung.

6. Ibu Dr. Eng. Ir. Mardiana, S.T., M.T., I.P.M., selaku Dosen Pembimbing Utama yang telah meluangkan banyak waktu, tenaga dan pikiran untuk memberikan bimbingan, arahan, serta motivasi yang luar biasa sejak awal hingga selesainya penelitian ini.
7. Bapak Rio Ariestia Pradipta, S.Kom., M.T.I., selaku Dosen Pembimbing Pendamping yang telah menjadi rekan diskusi teknis yang luar biasa, memberikan banyak ilmu, arahan teknis pengembangan sistem, serta kesabaran dalam membimbing dalam proses penyusunan.
8. Bapak Suyadi dan Seluruh Rekan Pengelola Suhita Bee Farm, yang telah memberikan izin observasi, menyediakan media penelitian serta bantuan informasi mengenai spesies lebah madu di lapangan.
9. Rekan tim penelitian, Ghebi Armando dan Leo Fetri Hendli, atas kerja sama yang solid, diskusi yang membangun, serta perjuangan bersama dalam menyelesaikan proyek BeeVra.
10. Seluruh responden yang telah meluangkan waktu untuk berpartisipasi dalam pengujian sistem sehingga diperoleh data yang objektif bagi penelitian ini.
11. Kedua orang tua dan keluarga besar terkasih, atas doa yang tiada henti, kasih sayang dan segala dukungan moril maupun materiil yang menjadi kekuatan utama dalam menyelesaikan masa studi.
12. Akbar, Dinda, Fani, Dela, Wulan dan Linda, sahabat-sahabat sejak masa SMP yang tetap setia menjaga tali persaudaraan ini. Terima kasih telah menjadi saksi pertumbuhan dan selalu ada untuk berbagi cerita melampaui waktu.
13. Ando, Ghebi, Melda, Nia dan Malvin, sahabat kuliah sekaligus teman seperjuangan. Terima kasih telah menjadi tempat bercerita, bertukar pikiran dan membuat dinamika perkuliahan menjadi lebih jauh menyenangkan dan bermakna.
14. Rekan-rekan Asisten Laboratorium Teknik Digital, atas pengalaman berharga, kerja sama, dan setiap momen bermakna di laboratorium yang telah membantu mendewasakan pola pikir teknis serta memberikan warna tersendiri dalam perjalanan akademik ini.

15. Basecamp Ivan, rumah ke sekian sebagai tempat melepas penat dan mencari inspirasi. Terima kasih atas canda tawa, kopi penyemangat malam, dan energi yang senantiasa menjaga api semangat perkuliahan ini tetap menyala.
16. Teman-teman seperjuangan B Bawa Bantal, atas segala tawa di ruang kelas, solidaritas yang kuat, dan memori tak terlupakan yang telah kita ukir bersama.
17. Keluarga Besar Rectical 22 atas kebersamaan, kekompakan dan persaudaraan yang telah terukir indah selama ini.

Penulis menyadari bahwa skripsi ini masih jauh dari kata sempurna. Oleh karena itu, kritik dan saran yang membangun sangat diharapkan. Semoga hasil penelitian ini dapat membawa keberkahan dan memberikan manfaat yang luas bagi perkembangan ilmu pengetahuan.

Bandar Lampung, 04 Juni 2026



Arswendo Erza Sadewa

NPM 2215061062

DAFTAR ISI

	Halaman
DAFTAR ISI	i
DAFTAR GAMBAR	iv
DAFTAR TABEL	viii
I. PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah.....	3
1.4 Tujuan Penelitian.....	4
1.5 Manfaat Penelitian	4
1.6 Sistematika Penulisan.....	5
II. TINJAUAN PUSTAKA	7
2.1 <i>User Interface</i>	7
2.2 <i>User Centered Design</i>	9
2.3 <i>Unified Modeling Language (UML)</i>	11
2.4 <i>Use Case Diagram</i>	11
2.5 <i>Activity Diagram</i>	12
2.6 <i>Frontend</i>	13
2.7 <i>Framework dan Library</i>	14
2.8 <i>React.js</i>	15
2.9 <i>Next.js</i>	16
2.10 <i>Shadcn UI</i>	18
2.11 <i>Metode Rapid Application Development (RAD)</i>	18
2.12 <i>Pengujian Perangkat Lunak</i>	20

2.13	<i>Black Box Testing</i>	21
2.14	Google Lighthouse	22
2.15	<i>System Usability Scale (SUS)</i>	22
2.16	Penelitian Terdahulu	24
III.	METODE PENELITIAN	30
3.1	Waktu dan Tempat Penelitian	30
3.2	Alat dan Bahan Penelitian	30
3.3	Tahapan Penelitian	32
3.3.1	Perencanaan Kebutuhan	34
3.3.2	Desain Sistem	35
3.3.3	Pengembangan	36
3.3.4	Implementasi	36
3.3.5	Penulisan Laporan	38
IV.	HASIL DAN PEMBAHASAN	39
4.1	Perencanaan Kebutuhan	39
4.1.1	<i>Understand Context of Use</i>	39
4.1.2	<i>Specify User Requirements</i>	41
4.2	Desain Sistem	44
4.2.1	<i>Design Solutions</i>	44
4.2.2	<i>Evaluate Against Requirements</i>	78
4.3	Pengembangan	79
4.3.1	Struktur Folder	79
4.3.2	Alur Kerja Sistem	80
4.3.3	Implementasi <i>Layout</i> dan Halaman Sistem	82
4.4	Implementasi	104
4.4.1	<i>Deployment</i>	104
4.4.2	Pengujian	106
4.5	Hasil dan Evaluasi	112
V.	KESIMPULAN DAN SARAN	115
5.1	Kesimpulan	115
5.2	Saran	116
	DAFTAR PUSTAKA	118

Lampiran A	Notulensi Observasi dan Evaluasi	123
Lampiran B	Desain	124
Lampiran C	Kode Program	126
Lampiran D	Hasil Pengujian <i>Black Box</i>	126
Lampiran E	Hasil Pengujian Performa	128
Lampiran F	Hasil Pengujian <i>Usability</i>	129
Lampiran G	Dokumentasi Kegiatan Penelitian.....	130

DAFTAR GAMBAR

Halaman

Gambar 1. Tahapan Metode UCD.....	9
Gambar 2. Logo React	16
Gambar 3. Tahapan Metode <i>Rapid Application Development</i>	19
Gambar 4. Ilustrasi <i>black box testing</i>	21
Gambar 5. Kategori Penilaian SUS.....	23
Gambar 6. Pembagian Pekerjaan Tim Penelitian.....	32
Gambar 7. Tahapan Penelitian	34
Gambar 8. <i>User Persona 1</i>	39
Gambar 9. <i>User Persona 2</i>	40
Gambar 10. <i>Use Case Diagram</i>	44
Gambar 11. <i>Activity Diagram Login</i>	46
Gambar 12. <i>Activity Diagram Logout</i>	47
Gambar 13. <i>Activity Diagram Registrasi</i>	48
Gambar 14. Lupa Kata Sandi	49
Gambar 15. Konfirmasi Akun Praktisi oleh Super Admin.....	50
Gambar 16, Menonaktifkan Akun Praktisi oleh Super Admin	51
Gambar 17. Mengaktifkan Kembali Akun Praktisi.....	52
Gambar 18. Edit Profil	53
Gambar 19. Ganti Kata Sandi	54
Gambar 20. Melihat Informasi Lebah.....	55
Gambar 21. Menambah Data Informasi Lebah.....	56
Gambar 22. Mengedit Data Informasi Lebah	57
Gambar 23. Menghapus Data informasi Lebah	58
Gambar 24. Klasifikasi Citra Lebah.....	59

Gambar 25. Riwayat Klasifikasi Lebah	60
Gambar 26. <i>Low Fidelity</i> Halaman <i>Login</i>	61
Gambar 27. <i>Low Fidelity</i> Halaman Daftar	61
Gambar 28. <i>Low Fidelity</i> Halaman Verifikasi Peran.....	61
Gambar 29. <i>Low Fidelity</i> Halaman OTP.....	62
Gambar 30. <i>Low Fidelity</i> Halaman <i>Reset Password</i>	62
Gambar 31. <i>Low Fidelity</i> Halaman Lupa Kata Sandi	62
Gambar 32. <i>Low Fidelity</i> Halaman Beranda.....	63
Gambar 33. <i>Low Fidelity</i> Halaman Informasi Lebah.....	63
Gambar 34. <i>Low Fidelity Form</i> Tambah Data	64
Gambar 35. <i>Low Fidelity</i> Detail Informasi	64
Gambar 36. <i>Low Fidelity Form</i> Edit Data	64
Gambar 37. <i>Low Fidelity</i> Konfirmasi Hapus Data.....	64
Gambar 38. <i>Low Fidelity</i> Halaman Klasifikasi Lebah.....	65
Gambar 39. <i>Low Fidelity</i> Detail Riwayat Klasifikasi	65
Gambar 40. <i>Low Fidelity</i> Halaman <i>Dashboard</i> Admin	65
Gambar 41. <i>Low Fidelity</i> Detail Akun Praktisi.....	66
Gambar 42. <i>Low Fidelity</i> Konfirmasi Permintaan Akun Praktisi	66
Gambar 43. <i>Low Fidelity</i> Konfirmasi Menonaktifkan Akun Praktisi	66
Gambar 44. <i>Low Fidelity</i> Konfirmasi Pengaktifan Kembali Akun Praktisi	66
Gambar 45. <i>Low Fidelity</i> Halaman Profil	67
Gambar 46. <i>Low Fidelity</i> Edit Profil.....	67
Gambar 47. <i>Low Fidelity</i> Konfirmasi <i>Logout</i>	67
Gambar 48. <i>Low Fidelity Form</i> Ganti Kata Sandi	68
Gambar 49. <i>High Fidelity</i> Halaman <i>Login</i>	68
Gambar 50. <i>High Fidelity</i> Halaman Daftar.....	69
Gambar 51. <i>High Fidelity</i> Halaman Verifikasi Peran	69
Gambar 52. <i>High Fidelity</i> Halaman OTP	69
Gambar 53. <i>High Fidelity</i> Halaman Lupa Kata Sandi	70
Gambar 54. <i>High Fidelity</i> Halaman Reset Password.....	70
Gambar 55. <i>High Fidelity</i> Halaman Beranda.....	71
Gambar 56. <i>High Fidelity</i> Halaman Informasi Lebah	72

Gambar 57. <i>High Fidelity</i> Form Tambah Data	72
Gambar 58. <i>High Fidelity</i> Detail Informasi Lebah	72
Gambar 59. <i>High Fidelity</i> Form Edit Data.....	73
Gambar 60. <i>High Fidelity</i> Konfirmasi Hapus Data Lebah	73
Gambar 61. <i>High Fidelity</i> Halaman Klasifikasi Lebah.....	73
Gambar 62. <i>High Fidelity</i> Detail Riwayat Klasifikasi	74
Gambar 63. <i>High Fidelity</i> Dashboard Admin	74
Gambar 64. <i>High Fidelity</i> Detail Akun Praktisi.....	75
Gambar 65. <i>High Fidelity</i> Konfirmasi Penolakan Akun Praktisi.....	75
Gambar 66. <i>High Fidelity</i> Konfirmasi Penonaktifan Akun Praktisi	75
Gambar 67. <i>High Fidelity</i> Konfirmasi Pengaktifan Kembali Akun Praktis.....	76
Gambar 68. <i>High Fidelity</i> Halaman Profil.....	76
Gambar 69. <i>High Fidelity</i> Edit Akun Pengguna	77
Gambar 70. <i>High Fidelity</i> Konfirmasi <i>Logout</i>	77
Gambar 71. <i>High Fidelity</i> Form Input Ganti Kata Sandi.....	77
Gambar 72. Struktur Folder	79
Gambar 73. Diagram Alur Kerja Sistem	80
Gambar 74. Arsitektur Sistem.....	104
Gambar 75. <i>Dashboard Deployment Website</i> BeeVra di Vercel	105
Gambar 76. Hasil Pengujian Performa Halaman Beranda.....	110
Gambar 78. Notulensi Observasi	123
Gambar 79. Notulensi Evaluasi.....	123
Gambar 80. Kode QR Notulensi	124
Gambar 81. <i>Low Fidelity Modal Form</i> Edit Data Pengguna	124
Gambar 82. <i>High Fidelity Modal Form</i> Edit Data Pengguna	125
Gambar 83. Kode QR Desain Figma	125
Gambar 84. Kode QR <i>Repository</i> Github	126
Gambar 85. Kode QR Hasil Pengujian <i>Black Box</i>	127
Gambar 86. Hasil Pengujian Halaman Klasifikasi Lebah.....	128
Gambar 87. Hasil Pengujian Halaman Informasi Lebah	128
Gambar 88. Kode QR Hasil Pengujian Performa	129
Gambar 89. Kode QR Hasil Pengujian <i>Usability</i>	130

Gambar 90. Dokumentasi Setelah Observasi.....	130
Gambar 91. Dokumentasi Setelah Pengujian.....	131
Gambar 92. Kode QR Dokumentasi Kegiatan Penelitian.....	131

DAFTAR TABEL

	Halaman
Tabel 1 Notasi <i>Use Case</i> Diagram	12
Tabel 2 Notasi <i>Activity</i> Diagram	13
Tabel 3 Perbandingan <i>Framework</i> dan <i>Library</i>	15
Tabel 4 Tahapan Metode <i>Rapid Application Development</i>	19
Tabel 5 Jadwal Kegiatan Penelitian	30
Tabel 6 Alat Penelitian (<i>Hardware</i>)	31
Tabel 7 Alat Penelitian (<i>Software</i>).....	31
Tabel 8 Bahan Penelitian.....	32
Tabel 9 Tahapan Perencanaan Kebutuhan.....	34
Tabel 10 Tahapan Desain Sistem	35
Tabel 11 Tahapan Pengembangan	36
Tabel 12 Tahapan Implementasi.....	37
Tabel 13 <i>Pain Points</i>	40
Tabel 14 Kebutuhan Fungsional.....	41
Tabel 15 Kebutuhan Non Fungsional.....	43
Tabel 16 Evaluasi	78
Tabel 17 <i>Black Box Testing</i> Fitur Utama Sistem.....	106
Tabel 18 Hasil Pengujian Performa.....	110
Tabel 19 Hasil Pengujian SUS	111
Tabel 20 Hasil Pengujian Black Box	126
Tabel 21 Lampiran Hasil Pengujian SUS.....	129

I. PENDAHULUAN

1.1 Latar Belakang

Indonesia dikenal sebagai negara dengan kekayaan flora dan fauna yang melimpah, termasuk sebagai pusat keanekaragaman hayati lebah madu asli yang paling beragam di dunia. Keragaman ini mencakup berbagai spesies dalam genus *Apis*, seperti *Apis dorsata*, *Apis cerana*, hingga *Apis mellifera*, serta berbagai spesies dalam genus *Trigona* sp. Tingginya jumlah spesies dan sub spesies yang tersebar di berbagai wilayah menunjukkan betapa kompleksnya ekosistem lebah di kawasan Indo-Malaya. *Apis mellifera* merupakan salah satu jenis lebah unggul yang banyak ditenakkan karena sifatnya jinak dan mampu menghasilkan madu dalam jumlah cukup banyak [1]. Selain itu, lebah ini memiliki banyak sub spesies dengan karakteristik dan kualitas madu yang berbeda-beda, sehingga proses identifikasi dan klasifikasi sub spesies lebah madu menjadi penting untuk pengelolaan budidaya yang lebih optimal.

Keragaman yang tinggi ini membawa tantangan yang signifikan dalam hal identifikasi. Bagi para peneliti, peternak dan pegiat konservasi, kemampuan untuk membedakan satu spesies dari yang lain secara akurat adalah hal yang krusial. Identifikasi yang tepat sangat penting untuk program pembiakan serta upaya konservasi. Secara tradisional, identifikasi bergantung pada metode morfometri yang menganalisis karakter fisik lebah. Namun, pendekatan ini memiliki kelemahan seperti prosesnya lambat, memerlukan keahlian taksonomi yang mendalam dan tingkat presisi yang tidak selalu konsisten.

Seiring dengan kemajuan teknologi informasi, berbagai penelitian mulai memanfaatkan kecerdasan buatan atau *Artificial Intelligence* untuk membantu proses identifikasi spesies secara otomatis. Salah satu metode yang terbukti unggul dalam mengenali pola citra digital adalah *Convolutional Neural Network* (CNN). Penelitian terdahulu menunjukkan bahwa metode CNN mampu melakukan klasifikasi spesies lebah madu dengan akurasi tinggi, bahkan mengungguli metode morfometri tradisional [2].

Meskipun model CNN telah menunjukkan kinerja yang sangat baik dalam klasifikasi citra, efektivitasnya dalam praktik sehari-hari akan optimal jika didukung oleh antarmuka yang mudah diakses dan interaktif bagi pengguna. Banyak penelitian masih berfokus pada pengembangan dan evaluasi model itu sendiri, tanpa menyediakan sarana yang praktis bagi peneliti, peternak atau masyarakat umum untuk memanfaatkan teknologi ini secara langsung. Di sinilah pengembangan *frontend website* menjadi sangat krusial. Sebuah *website* akan berfungsi sebagai jembatan yang menghubungkan kemampuan prediksi kompleks dari model CNN di *backend* dengan kebutuhan interaksi pengguna di *frontend*. Antarmuka *website* ini menerima *input* citra, mengirimkan ke model dan menyajikan hasil klasifikasi secara informatif, responsif dan menarik, sehingga pengetahuan ilmiah dapat diakses dan dimanfaatkan secara praktis.

Metode *Rapid Application Development* (RAD) dipilih sebagai pendekatan untuk membangun *frontend* yang efisien, adaptif dan berpusat pada pengguna karena sifat dari metode RAD yang cepat, iteratif, dan sangat menekankan keterlibatan pengguna dalam setiap tahap. Pendekatan RAD memungkinkan pengembangan prototipe secara progresif, mengumpulkan *feedback* awal, dan melakukan penyesuaian desain secara cepat untuk memastikan produk akhir benar-benar memenuhi kebutuhan pengguna. Dalam proses pengembangannya, *framework* Next.js berbasis React.js digunakan untuk membangun tampilan *frontend* yang modern, dinamis dan mudah dikelola. Kombinasi Next.js dengan metode RAD diharapkan dapat mempercepat proses pengembangan sekaligus menghasilkan tampilan antarmuka yang efisien dan adaptif di berbagai perangkat.

Dengan demikian, penelitian ini berfokus pada pengembangan *frontend website* klasifikasi spesies lebah madu Indo-Malaya berbasis metode *Rapid Application Development* (RAD). Melalui penelitian ini, diharapkan dapat dihasilkan sistem yang tidak hanya berfungsi dengan baik secara teknis dalam menyajikan hasil klasifikasi, tetapi juga memiliki tingkat kemudahan penggunaan (*usability*) yang tinggi dan memberikan pengalaman interaksi yang intuitif bagi pengguna.

1.2 Rumusan Masalah

Berdasarkan uraian latar belakang di atas, terdapat beberapa permasalahan yang perlu dijawab dalam penelitian ini:

1. Bagaimana proses perancangan dan pembangunan *frontend website* sistem klasifikasi spesies lebah madu Indo-Malaya melalui penerapan metode *Rapid Application Development* (RAD)?
2. Bagaimana prinsip-prinsip *User Centered Design* (UCD) diterapkan dalam merancang antarmuka dan mengimplementasikan antarmuka *website* yang interaktif, informatif dan mudah digunakan?
3. Bagaimana hasil pengujian fungsionalitas, performa dan *usability* dari *frontend* yang dikembangkan menggunakan metode RAD?

1.3 Batasan Masalah

Agar penelitian ini lebih terfokus dan terarah, serta sesuai dengan tujuan yang ingin dicapai, maka ditetapkan batasan masalah sebagai berikut:

1. Penelitian ini berfokus pada perancangan dan implementasi antarmuka *website*, sedangkan aspek *backend* seperti pembuatan API, logika server dan manajemen basis data tidak termasuk dalam lingkup penelitian ini.
2. Model klasifikasi CNN yang digunakan dianggap sudah tersedia dan diakses melalui RESTful API, sehingga penelitian tidak membahas proses pelatihan model.

3. Pengujian sistem hanya difokuskan pada aspek *frontend*, meliputi fungsionalitas, *usability*, responsivitas tampilan dan performa *client side* tanpa menilai keamanan sisi server.

1.4 Tujuan Penelitian

Berdasarkan rumusan masalah yang telah dipaparkan, maka tujuan yang ingin dicapai melalui penelitian ini adalah sebagai berikut:

1. Merancang dan membangun *frontend website* untuk sistem klasifikasi spesies lebah madu Indo-Malaya dengan menerapkan metode *Rapid Application Development* (RAD).
2. Merancang dan mengimplementasikan antarmuka *website* yang informatif dengan prinsip-prinsip *User Centered Design* (UCD) untuk memfasilitasi pengguna dalam mendapatkan informasi dan melakukan klasifikasi spesies lebah madu berbasis CNN.
3. Melakukan pengujian fungsionalitas, performa dan *usability* untuk mengevaluasi kinerja dan pengalaman pengguna terhadap tampilan *frontend* yang dikembangkan.

1.5 Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan beberapa manfaat sebagai berikut:

1. Dalam hal akademis, penelitian ini memberikan kontribusi terhadap pengembangan ilmu rekayasa perangkat lunak, khususnya pada penerapan metode *Rapid Application Development* (RAD) yang terintegrasi dengan prinsip-prinsip *User Centered Design* (UCD) dalam pengembangan *frontend* berbasis React.js.
2. Dari sisi praktis, penelitian ini dapat membantu pengguna seperti peneliti atau peternak lebah dalam mengidentifikasi spesies lebah madu secara lebih cepat dan akurat tanpa memerlukan keahlian morfologi yang mendalam.

3. Secara teknologi, penelitian ini menunjukkan teknologi modern *frontend* seperti React.js, Shadcn UI dan integrasi RESTful API dalam membangun sistem web berbasis kecerdasan buatan yang adaptif dan responsif.

1.6 Sistematika Penulisan

Untuk memudahkan pemahaman terhadap isi penelitian ini, sistematika penulisan laporan disusun sebagai berikut:

a) BAB I PENDAHULUAN

Bab ini menguraikan latar belakang masalah yang melandasi penelitian, perumusan masalah yang akan dijawab, batasan masalah untuk menjaga fokus penelitian, tujuan yang ingin dicapai, manfaat yang diharapkan, serta diakhiri dengan sistematika penulisan laporan.

b) BAB II TINJAUAN PUSTAKA

Bab ini membahas berbagai teori dan konsep yang relevan dengan penelitian, meliputi konsep *frontend*, *User Interface*, *User Centered Design* (UCD), *Unified Modeling Language* (UML) dengan *use case* dan *activity diagram*, *framework* dan *library* termasuk React.js, Next.js dan Shadcn UI, metode *Rapid Application Development* (RAD), pengujian perangkat lunak secara fungsional (*black box testing*), pengujian performa dengan Google Lighthouse dan pengujian kegunaan dengan menggunakan metode *System Usability Scale* (SUS).

c) BAB III METODOLOGI PENELITIAN

Bab ini menjelaskan pendekatan dan tahapan penelitian yang dilakukan. Pembahasan meliputi waktu dan tempat penelitian, alat dan bahan yang digunakan, pembagian pekerjaan dalam tim, serta tahapan penelitian yang mengacu pada metode *Rapid Application Development* (RAD), meliputi perencanaan, desain sistem, pengembangan dan implementasi.

d) BAB IV HASIL DAN PEMBAHASAN

Bab ini menyajikan hasil seluruh proses penelitian, dimulai dari hasil analisis kebutuhan, perancangan sistem, implementasi hingga hasil

pengujian yang meliputi pengujian fungsionalitas, performa dan pengujian *usability*.

e) **BAB V KESIMPULAN DAN SARAN**

Bab ini berisi rangkuman dari seluruh hasil penelitian sebagai jawaban atas tujuan penelitian, serta memberikan saran dan rekomendasi untuk pengembangan sistem lebih lanjut di masa mendatang.

II. TINJAUAN PUSTAKA

2.1 *User Interface*

User Interface atau antarmuka pengguna merupakan segala sesuatu yang dapat dilihat dan disentuh oleh pengguna ketika menggunakan produk digital, mulai dari tata letak, warna, tipografi, hingga elemen interaktif seperti tombol dan ikon. Tujuan utama dari UI adalah menciptakan antarmuka yang tidak hanya menarik secara visual, tetapi juga mudah dipahami dan digunakan. Secara sederhana, *User Interface* merupakan titik pertemuan antara manusia dengan mesin, pengguna dapat memberikan perintah dan menerima umpan balik dari sistem[3].

Interaksi pengguna dengan UI umumnya terjadi melalui beberapa jenis antarmuka, seperti *Graphical User Interface* (GUI) yang merupakan antarmuka berbasis visual yang menampilkan ikon, tombol dan menu navigasi, lalu *Voice Controlled Interface* (VUI) yang memanfaatkan suara untuk berinteraksi dengan sistem dan *Gesture Based Interface* yang menggunakan gerakan tubuh atau sentuhan sebagai *input* interaktif [4].

Dalam merancang UI, diperlukan prinsip-prinsip desain yang tepat agar sistem mudah digunakan dan memberikan pengalaman yang menyenangkan. Beberapa prinsip utama dalam *User Interface Design* adalah:

a. Konsistensi (*Consistency*)

Konsistensi berperan penting dalam membantu pengguna memahami pola dan cara kerja antarmuka. Elemen desain yang seragam, baik dari segi warna,

huruf, ikon, maupun tata letak akan memudahkan pengguna beradaptasi dengan tampilan.

b. Kesederhanaan dan Minimalisme (*Simplicity and Minimalism*)

Antarmuka yang sederhana membantu mengurangi kebingungan pengguna. Prinsip ini menekankan pentingnya menghilangkan elemen yang tidak memberikan nilai tambah, menjaga tata letak tetap bersih, serta memfokuskan pengguna pada tujuan utama.

c. Responsif dan Aksesibilitas (*Responsiveness and Accessibility*)

Antarmuka harus dapat beradaptasi dengan berbagai ukuran layar dan perangkat, baik *desktop* maupun *mobile*. Selain itu perlu mempertimbangkan aksesibilitas bagi pengguna dengan keterbatasan visual, pendengaran atau mobilitas, seperti penggunaan kontras warna yang memadai dan dukungan interaksi melalui keyboard.

d. Hirarki Visual (*Visual Hierarchy*)

Hirarki visual membantu pengguna mengenali informasi berdasarkan tingkat kepentingannya. Elemen penting harus menonjol melalui ukuran, warna atau penempatan strategis.

e. Umpan Balik yang Jelas (*Clear Feedback*)

Sistem perlu mempertimbangkan umpan balik yang cepat dan mudah dipahami setiap kali pengguna melakukan interaksi. Umpan balik dapat diterapkan sebagai perubahan warna, animasi, atau pesan status yang menandakan tindakan berhasil atau gagal. Dengan begitu pengguna memahami bahwa sistem merespons tindakannya dengan benar.

f. Fleksibilitas dan Efisiensi Penggunaan (*Flexibility and Efficiency of Use*)

Antarmuka yang fleksibel memungkinkan pengguna dari berbagai tingkat keahlian dapat menyelesaikan tugas dengan cepat. Penyediaan *shortcut* opsi

akses cepat, dan kemampuan menyesuaikan tampilan dapat meningkatkan efisiensi pada penggunaan sistemnya.

g. Pencegahan Kesalahan (*Preventing Errors*)

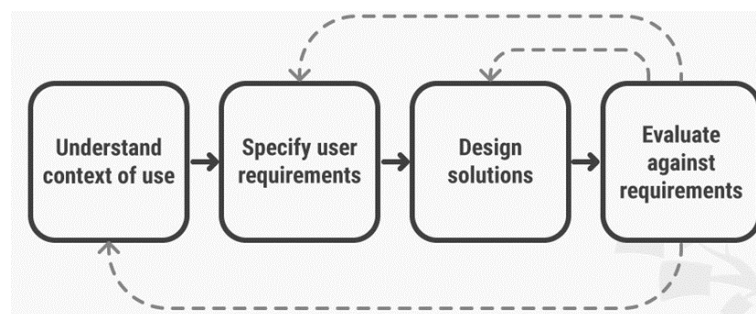
Desain antarmuka harus mampu mencegah terjadinya kesalahan dengan menyediakan validasi *input*, kontrol yang sesuai dan opsi *undo-redo*. Dengan begitu pengguna dapat menghindari kesalahan sebelum terjadi serta dapat memperbaiki jika diperlukan.

h. Desain Estetis dan Minimalis (*Aesthetic and Minimalist Design*)

Tampilan antarmuka harus bersih, sederhana dan tidak menampilkan elemen yang tidak diperlukan. Desain yang estetis membantu mengurangi beban kognitif pengguna serta meningkatkan kenyamanan dan fokus saat berinteraksi [5].

2.2 *User Centered Design*

User Centered Design (UCD) adalah pendekatan pengembangan antarmuka yang melibatkan pengguna dari tahap awal hingga akhir implementasi, sehingga menghasilkan produk yang sesuai dengan kebutuhan pasar. Konsep utama UCD adalah menempatkan pengguna sebagai pusat dari proses pengembangan sistem, yang berarti tujuan, sifat-sifat dan konteks sistem didasarkan pada pengalaman pengguna.



Gambar 1. Tahapan Metode UCD

Dalam pendekatan UCD, prinsip-prinsip yang harus diperhatikan yaitu adalah sebagai berikut:

a. *Understand Context of Use*

Tahap pertama dalam UCD adalah memahami konteks penggunaan aplikasi yang akan dibangun yang bertujuan untuk menggali informasi dari calon pengguna. Pada tahap ini terdapat empat poin pengerjaan, yaitu menentukan calon pengguna, wawancara untuk menggali informasi, membuat *user persona* dan menentukan *pain points*. Teknik wawancara yang digunakan untuk pengambilan data adalah semi terstruktur, yaitu menggunakan sejumlah daftar panduan pertanyaan yang kemudian dapat berkembang sesuai topik yang dibahas.

b. *Specify User Requirement*

Tahap ini memungkinkan pengguna untuk mencapai tujuannya ketika menggunakan sebuah sistem dan merupakan kunci dasar dari *user interface* dan *user experience*. Dengan memahami pengguna, tujuan dan konteksnya, maka sistem yang dibuat akan semakin berguna dan sesuai dengan kebutuhan. *Requirement* atau kebutuhan berisi tentang apa yang harus diberikan atau bagaimana pengguna berinteraksi dari suatu produk.

c. *Design Solutions*

Proses *design solutions* digunakan untuk menerjemahkan segala kebutuhan yang telah dianalisis yang berasal dari ekspektasi *stakeholder* ke dalam solusi desain. Desain dimulai dengan *low-fidelity* sebagai representasi desain awal yang sederhana dan kasar untuk menguji ide dan konsep. Selanjutnya desain diwujudkan secara detail dalam bentuk rancangan visual *high-fidelity*. Rancangan ini yang menjadi spesifikasi akhir dan acuan utama untuk tahap pengembangan dan menjadi dasar verifikasi pada tahap pengujian produk.

d. *Evaluate Against Requirements*

Tahap terakhir adalah melakukan evaluasi terhadap kebutuhan. Tahapan ini bertujuan untuk menilai desain secara langsung kepada pengguna agar dapat

diketahui tingkat kesesuaian dengan kebutuhan pengguna yang telah ditetapkan sebelumnya [6].

2.3 *Unified Modeling Language (UML)*

Unified Modeling Language (UML) merupakan bahasa pemodelan standar yang digunakan untuk menggambarkan, memvisualisasikan dan mendokumentasikan sistem perangkat lunak. UML berfungsi sebagai alat bantu dalam merancang serta memodelkan struktur dan perilaku sistem secara sistematis. Sebagai notasi grafis yang umum digunakan di bidang rekayasa perangkat lunak, UML membantu pengembang, *stakeholder* dan analis dalam memahami keterkaitan antar komponen sistem sebelum diimplementasikan. UML memiliki banyak jenis diagram, namun *Use Case* diagram dan *Activity* diagram merupakan diagram yang paling umum digunakan pada perancangan sistem[7].

2.4 *Use Case Diagram*

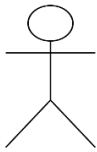
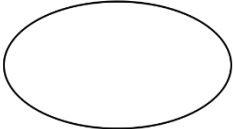
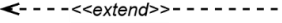
Use Case diagram merupakan salah satu jenis diagram pada UML yang digunakan untuk menggambarkan hubungan interaksi antara pengguna (aktor) dengan sistem yang dibangun. *Use case* diagram menunjukkan model fungsional sistem melalui serangkaian fungsi (*use case*) yang dapat diakses oleh aktor berdasarkan kebutuhan dan peran dalam sistem. Setiap aktor digambarkan sesuai akses dan fungsinya terhadap sistem, sehingga diagram ini menjadi representasi visual mengenai siapa saja yang terlibat serta fungsi apa saja yang dapat dijalankan oleh sistem.

Use case diagram memiliki kegunaan untuk menunjukkan rangkaian tindakan atau aktivitas dalam sistem secara berurutan dan menggambarkan proses bisnis berdasarkan susunan kegiatan dilakukan oleh aktor terhadap sistem. Selain itu *use case* diagram juga memberikan manfaat dalam proses pengembangan sistem, yaitu:

- a. Membantu melakukan verifikasi sistem

- b. Menggambarkan kebutuhan antarmuka (*Interface*) dari sistem yang dirancang
- c. Menentukan keterlibatan masing-masing aktor serta fungsi apa yang menjadi tanggung jawab mereka
- d. Menyampaikan aturan atau ketentuan yang dibutuhkan sistem tanpa membingungkan pengguna
- e. Mempermudah komunikasi antara pengguna dan pihak yang memahami sistem secara teknis (*expert system*) [8].

Tabel 1 Notasi *Use Case Diagram*

Simbol	Nama	Deskripsi
	Aktor	Mewakili peran sistem lain, orang atau alat ketika berkomunikasi dengan <i>Use case</i>
	<i>Use Case</i>	Abstraksi dan interaksi antara actor dengan sistem
	<i>Association</i>	Hubungan antara objek satu dengan yang lainnya
	<i>Include</i>	Menspesifikasikan bahwa <i>Use Case</i> sumber secara eksplisit
	<i>Extend</i>	Menspesifikasikan bahwa <i>Use Case</i> target memperluas perilaku <i>Use Case</i> sumber pada suatu titik yang diberikan
	Sistem	Menspesifikasikan paket yang menampilkan sistem secara terbatas[9]

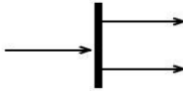
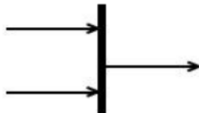
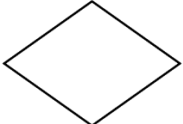
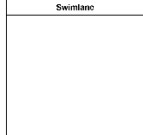
2.5 *Activity Diagram*

Activity diagram merupakan salah satu pemodelan yang digunakan dalam perancangan perangkat lunak untuk menggambarkan urutan aktivitas dalam suatu proses bisnis. Diagram ini membantu memahami alur proses sistem secara keseluruhan karena menampilkan aktivitas-aktivitas yang saling bergantung satu

dengan yang lain. Fokus utama *Activity* diagram adalah menggambarkan aktivitas dalam sistem, bukan menjelaskan apa yang dilakukan oleh aktor.

Activity diagram menunjukkan proses serta urutan aktivitas, termasuk adanya percabangan (*fork*), penggabungan (*join*) dan alur proses yang terjadi dalam sistem. *Activity* diagram disusun berdasarkan satu atau beberapa *use case* yang telah dibuat sebelumnya, yang artinya *activity* diagram adalah proses lanjut dari *use case* diagram[9].

Tabel 2 Notasi *Activity* Diagram

Simbol	Nama	Deskripsi
	<i>Initial</i>	Awal dari Kumpulan aktivitas atau aksi
	<i>Final</i>	Menghentikan seluruh aliran kontrol pada sebuah aksi atau aktivitas
	<i>Activities</i>	Kumpulan aktivitas atau aksi
	<i>Fork</i>	Memecah sebuah perilaku menjadi akses atau aktivitas paralel
	<i>Join</i>	Menggabungkan kembali menjadi satu aliran perilaku dari aktivitas paralel
	<i>Decision</i>	Suatu pengujian yang bertujuan untuk memastikan aliran kontrol dan aliran objek menuju menuju ke suatu arah
	<i>Swimlane</i>	Kolom yang berguna untuk mengelompokkan aktivitas tertentu berdasarkan actor

2.6 Frontend

Frontend adalah istilah yang merujuk pada bagian depan dari sebuah aplikasi atau situs web, yaitu bagian yang langsung dilihat dan berinteraksi dengan pengguna.

Pengembangan *frontend* berfokus pada pengalaman pengguna (*user experience*) dan mencakup semua kegiatan pengembangan antarmuka, termasuk pada tata letak, desain dan fungsionalitas interaktif. Secara teknis, *frontend* meliputi implementasi struktur dengan HTML, desain dan tata letak dengan CSS, serta logika interaktif dan pengelolaan sisi klien (browser) menggunakan JavaScript [10].

Dalam konteks klasifikasi lebah madu berbasis CNN, *frontend* berperan penting sebagai media penyajian hasil pemrosesan mode kecerdasan buatan kepada pengguna. Melalui antarmuka *website*, pengguna dapat mengunggah citra lebah madu untuk di klasifikasikan, kemudian menerima hasil prediksi yang ditampilkan secara visual dan informatif. Frontend menghubungkan hasil pemrosesan model di sisi *backend* dengan *end user* sehingga hasil klasifikasi yang dihasilkan oleh CNN dapat disajikan kepada *end user*.

2.7 *Framework dan Library*

Framework dan *library* merupakan dua elemen penting dalam pengembangan *software* modern, terutama pada pengembangan aplikasi berbasis web. *Library* adalah sekumpulan fungsi, metode atau utilitas yang dapat digunakan kembali untuk menyelesaikan tugas tertentu dalam proses pengembangan. *Library* memberikan kebebasan penuh kepada pengembang untuk menentukan bagaimana dan kapan fungsi-fungsi tersebut akan digunakan sesuai dengan kebutuhan proyek. Sementara itu, *framework* adalah kerangka kerja yang lebih komprehensif dan umumnya menentukan struktur kerja aplikasi yang harus diikuti oleh pengembang. *Framework* menyediakan aturan dan pola pengembangan yang lebih terstruktur sehingga mampu mengatur bagaimana kode ditulis, dijalankan dan diorganisasi di dalam proyek [10].

Meskipun keduanya memiliki tujuan yang sama, yaitu mempermudah dan mempercepat proses pengembangan, *framework* dan *library* memiliki perbedaan yang mendasar seperti yang disajikan pada tabel berikut:

Tabel 3 Perbandingan *Framework* dan *Library*

Aspek Perbandingan	<i>Framework</i>	<i>Library</i>
Kontrol	Mengendalikan alur kerja aplikasi dan memanggil kode yang telah dibuat oleh developer di tempat yang sesuai.	Pengembang memiliki kendali penuh dan dapat memanggil fungsi <i>library</i> sesuai kebutuhan.
Skala dan Kompleksitas	Umumnya lebih besar dan kompleks karena menyediakan solusi menyeluruh seperti <i>routing</i> , <i>state management</i> , <i>build</i> dan <i>deployment</i> .	Lebih ringan dan spesifik pada fungsi tertentu tanpa mengatur keseluruhan struktur aplikasi.
Tingkat Abstraksi	Menyediakan tingkat abstraksi tinggi dan menyembunyikan detail implementasi teknik.	Memberikan fleksibilitas lebih besar dan memungkinkan pengembang mengatur detail penggunaannya.
Tujuan Utama	Menyediakan kerangka kerja yang lengkap untuk membangun aplikasi dari awal hingga akhir.	Menyediakan alat bantu atau fungsi spesifik untuk menyelesaikan tugas tertentu.

2.8 React.js

React merupakan salah satu *library* JavaScript paling populer didunia yang digunakan untuk membangun antarmuka pengguna. React dikembangkan oleh Facebook dan pertama kali dirilis pada 2013. Sejak saat itu, React menjadi fondasi utama bagi banyak aplikasi web modern karena dapat digunakan untuk membangun antarmuka yang dinamis, efisien dan mudah di *maintenance*.

Salah satu konsep dalam React adalah penggunaan komponen (*components*) yang blok antarmuka yang bersifat mandiri dan dapat digunakan kembali. Setiap komponen mengelola logika dan *state*-nya sendiri sehingga pengembangan UI menjadi lebih modular dan terstruktur. Komponen dapat dibuat dalam dua bentuk, yaitu *Functional Components* yang didefinisikan sebagai fungsi JavaScript yang menerima prop dan mengembalikan elemen React, serta *Class Components* yang didefinisikan sebagai kelas ES6 yang mendukung *lifecycle methods*. Selain itu,

React menggunakan JSX (JavaScript XML), yaitu sintaks yang memungkinkan penulisan kode mirip HTML di dalam JavaScript sehingga tampilan UI dapat didefinisikan dengan cara yang lebih deklaratif dan mudah dipahami.



Gambar 2. Logo React

React juga menggunakan dua konsep penting dalam pengelolaan data, yaitu *State* yang merupakan objek internal dalam komponen yang menyimpan data yang dapat berubah selama siklus hidup komponen, ketika *state* mengalami perubahan, React akan secara otomatis melakukan *re-render* untuk memperbarui tampilan agar sesuai dengan data terbaru. Sedangkan *Props* digunakan untuk mengirimkan data dari komponen induk ke *children*. *Props* juga bersifat *ready-only* yang artinya komponen *children* tidak dapat mengubah nilai *props* yang diterima.

Salah satu inovasi terpenting yang dibawa oleh React adalah Virtual DOM yang merupakan representasi memori dari DOM asli. Ketika *state* atau *props* diubah maka React tidak langsung memperbarui DOM di browser, melainkan memperbarui VDOM terlebih dahulu, kemudian membandingkan dengan versi sebelumnya melalui proses yang disebut *reconcillation*. Hanya perubahan yang akan diterapkan di DOM asli sehingga proses pembaruan antarmuka menjadi lebih cepat dan efisien [10].

2.9 Next.js

Next.js merupakan kerangka kerja atau *framework* berbasis React yang dikembangkan oleh Vercel untuk menyelesaikan berbagai permasalahan umum

dalam pengembangan React murni, seperti *server-side rendering* (SSR), *static site generation* (SSG) dan sistem *routing* yang kompleks. *Framework* ini secara otomatis mengelola berbagai aspek teknis seperti optimasi performa dan SEO sehingga pengembang dapat berfokus pada pembuatan fitur utama aplikasi. Melalui kombinasi fitur SSR dan SSG, Next.js mampu meningkatkan performa dan efisiensi SEO dengan menghasilkan halaman yang dapat di-*render* di sisi server maupun statis.

Selain itu, Next.js menghadirkan sejumlah keunggulan yang tidak dimiliki React murni. *Framework* ini menyediakan *Smart Routing*, yaitu sistem perutean halaman dilakukan secara otomatis melalui struktur folder *pages*, sehingga pengembang tidak perlu melakukan konfigurasi manual atau menggunakan pustaka tambahan seperti React Router. Next.js juga mendukung optimasi gambar dari aset secara bawaan melalui komponen *next/image*, yang memungkinkan proses pemuatan gambar lebih efisien tanpa menurunkan kualitas tampilan sehingga berdampak pada peningkatan kinerja aplikasi serta pengalaman pengguna yang lebih baik, dikarenakan halaman akan terasa lebih ringan dan responsif. Selain itu juga terdapat fitur *Automatic Code Splitting* yang memungkinkan Next.js memecah kode aplikasi menjadi bagian-bagian kecil (*chunks*) yang dimuat sesuai kebutuhan, sehingga dengan mekanisme ini maka waktu muat halaman akan lebih cepat dan mengurangi beban pada sisi klien [11].

Dalam pengembangan, file komponen pada Next.js umumnya menggunakan ekstensi *.tsx*, karena *framework* ini mendukung penggunaan TypeScript secara bawaan. TypeScript merupakan superset dari JavaScript yang menyediakan fitur tipe data statis dan peningkatan lain dalam proses pengembangan, serta membantu pengguna untuk mengidentifikasi kesalahan kode sejak tahap penulisan (*compile time*), bukan setelah aplikasi dijalankan (*runtime*) [12].

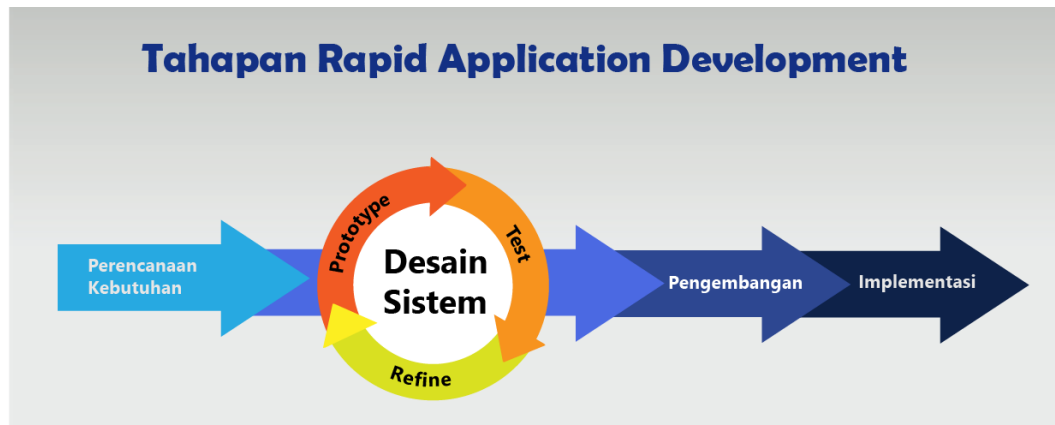
2.10 Shadcn UI

Shadcn UI merupakan kumpulan komponen antarmuka modern berbasis React yang bersifat open source. Berbeda dengan pustaka UI yang lain, Shadcn tidak di instal melalui *npm*, melainkan dengan menyalin kode komponen secara langsung ke dalam proyek. Disediakan perintah CLI yang memudahkan proses penyalinan komponen ke dalam direktori “src/components/ui”, sehingga proyek lebih ringan karena hanya berisi komponen yang diperlukan dan digunakan. Selain itu setiap komponen Shadcn juga menggunakan Tailwind CSS dalam pengaturan tampilannya sehingga menghasilkan desain yang seragam, modern, serta responsif di berbagai ukuran layar [13].

Shadcn mendukung integrasi dengan *React Hook Form* dan *zod* melalui komponen *form* yang memudahkan proses validasi data pada formulir secara efisien dan konsisten. Struktur komponen Shadcn juga bersifat modular, yang artinya setiap bagian antarmuka dikembangkan secara terpisah namun tetap digunakan bersama dengan mudah. Oleh karena itu, Shadcn dipilih untuk membantu dalam pengembangan yang didasarkan pada ketersediaan komponen siap pakai dengan tampilan yang modern dan konsisten serta kemudahan integrasi dengan React.js sehingga proses pengembangan *frontend* menjadi lebih cepat dan efisien.

2.11 Metode *Rapid Application Development* (RAD)

Metode *Rapid Application Development* (RAD) adalah metode pengembangan perangkat lunak yang menekankan kecepatan dalam siklus pengembangan. Pendekatan ini berfokus pada pembuatan sistem secara iteratif dan interaktif dengan melibatkan pengguna secara langsung dalam setiap tahapnya. RAD menggunakan pendekatan berorientasi objek dan menekankan pada pembangunan prototipe yang cepat agar sistem dapat segera diuji dan disesuaikan dengan kebutuhan pengguna [14][15].



Gambar 3. Tahapan Metode *Rapid Application Development*

Metode pengembangan RAD memiliki empat tahapan utama yang digunakan dalam proses pengembangannya, yaitu sebagai berikut:

Tabel 4 Tahapan Metode *Rapid Application Development*

Tahapan	Deskripsi
Perancangan Kebutuhan	Tahap awal untuk mengidentifikasi masalah dan pengumpulan data dari <i>stakeholder</i> atau pengguna, serta menentukan tujuan dan kebutuhan sistem. Keterlibatan pengguna sangat penting untuk memastikan sistem yang dikembangkan sesuai dengan harapan dan kebutuhan yang diinginkan.
Desain Sistem	Tahap ini berfokus pada perancangan struktur sistem berdasarkan kebutuhan yang telah diidentifikasi. Proses desain dilakukan secara interaktif dan berulang untuk memperbaiki ketidaksesuaian yang mungkin ditemukan. Hasil dari tahap ini meliputi rancangan <i>software</i> yang mencakup struktur data, organisasi sistem secara keseluruhan dan elemen lainnya.
Pengembangan	Pada tahap ini, desain yang telah disetujui diubah menjadi aplikasi versi beta (<i>prototype</i>). Pengembang terus melakukan pengembangan, integrasi dan penyempurnaan berdasarkan <i>feedback</i> dari pengguna hingga aplikasi memenuhi kebutuhan yang telah ditetapkan.
Implementasi	Pada tahap akhir ini, sistem yang telah dikembangkan akan diuji dan diterapkan. Pengujian dilakukan untuk memastikan sistem dapat berjalan sesuai fungsinya dan bebas dari kesalahan. Setelah diuji dan mendapatkan hasil yang baik maka sistem akan disetujui dan dapat digunakan secara resmi oleh pengguna.

Metode RAD dipilih karena mendukung pengembangan antarmuka *website* yang membutuhkan waktu singkat namun tetap responsif terhadap kebutuhan pengguna. Pendekatan interaktif yang diterapkan pada RAD memungkinkan pengembang untuk membuat *prototype* secara cepat, kemudian menguji dan memperbaikinya berdasarkan *feedback* secara langsung oleh pengguna. Hal ini sangat penting dalam proyek sistem klasifikasi spesies lebah madu karena tampilan hasil prediksi dari model CNN perlu disesuaikan agar mudah dipahami pengguna dan memberikan pengalaman interaksi yang intuitif bagi pengguna.

2.12 Pengujian Perangkat Lunak

Pengujian perangkat lunak atau *testing* merupakan tahapan yang dilakukan setelah proses implementasi pengkodean selesai. Pengujian bertujuan untuk menemukan dan mendeteksi kesalahan (*error*) atau cacat (*bug*) pada perangkat lunak. Dengan melakukan pengujian, dapat diketahui apakah perangkat lunak sudah berfungsi sesuai dengan spesifikasi dan kebutuhan fungsi yang telah ditentukan.

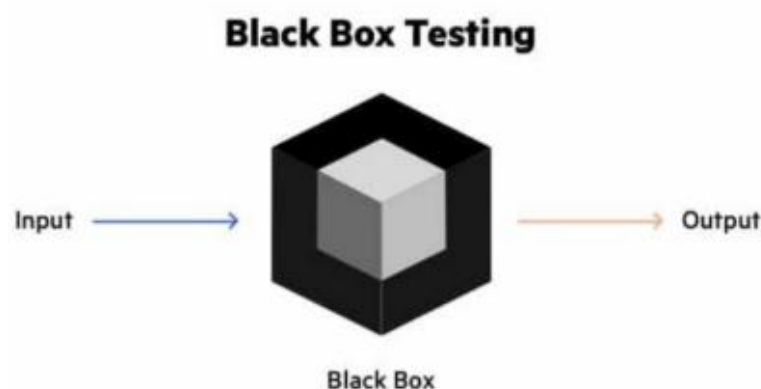
Pengujian perangkat lunak dapat dilakukan secara manual dan otomatis. Pada pengujian manual, penguji menyiapkan *test case* dan menjalankan pengujian secara langsung untuk mengetahui *error* dalam sistem. Sedangkan pada pengujian otomatis digunakan teknik *scripting* atau *tools* untuk mengeksekusi pengujian secara otomatis sesuai hasil yang diharapkan [16].

Secara umum, pengujian perangkat lunak mencakup berbagai aspek. Tiga aspek penting yang menjadi fokus dalam konteks ini adalah *functional testing*, *performance testing* dan *usability testing*. *Functional testing* adalah jenis pengujian perangkat lunak yang bertujuan untuk memastikan bahwa perangkat lunak berfungsi dengan benar sesuai spesifikasi yang telah ditentukan. Tujuan utama dari pengujian fungsional adalah untuk menguji setiap fungsi perangkat lunak, mulai dari *input* hingga *output*, untuk memverifikasi bahwa fungsi-fungsi tersebut berjalan sesuai dengan persyaratan yang telah ditetapkan [17]. Selanjutnya,

pengujian performa atau *performance testing* bertujuan untuk mengevaluasi dengan mengukur kecepatan, responsivitas dan stabilitas sistem. Pengujian ini berfokus pada waktu muat halaman dan optimasi aset untuk memastikan apakah aplikasi berjalan ringan dan cepat saat diakses. Sedangkan, *usability testing* adalah sebuah proses untuk mengukur karakteristik interaksi antara pengguna dan sistem. Tujuan dari pengujian ini adalah untuk mengidentifikasi kelemahan-kelemahan pada antarmuka, sehingga perancangan sistem dapat diperbaiki dengan cepat dan tepat [18].

2.13 *Black Box Testing*

Black box testing merupakan metode pengujian yang digunakan untuk menguji fungsionalitas tanpa harus mengetahui struktur internal kode atau isi dari program tersebut. Prinsip dasar dari metode ini adalah memandang sistem sebagai kotak hitam (*black box*) yang hanya berfokus pada pengujian *input* dan *output* pada tampilan suatu aplikasi. Dalam pengujian ini, penguji mengetahui apa yang harus dilakukan sistem, namun tidak memiliki pengetahuan tentang bagaimana program melakukannya secara internal. Pengujian hanya perlu mengetahui apa peran sistem tersebut dan fungsinya, namun tidak mengetahui mekanisme kode internalnya



Gambar 4. Ilustrasi *black box testing*

Pengujian *black box* didasarkan pada detail aplikasi seperti tampilan, fungsi yang ada dan kesesuaian alur fungsi dengan proses bisnis yang diinginkan. Penguji tidak

melihat atau menguji *source code* program, namun hanya terfokus pada cara dan hasil dari informasi fungsi-fungsi yang ada dalam aplikasi [19].

2.14 Google Lighthouse

Google Lighthouse adalah alat audit *open source* yang dikembangkan oleh Google untuk mengevaluasi dan memberikan informasi bagi pengembang atau pemilik situs web mengenai performa, aksesibilitas, SEO dan *best practices* dari suatu situs web. Meskipun alat ini menyediakan fitur audit untuk aksesibilitas, SEO dan PWA, fungsi utamanya dalam testing yang dilakukan pada proyek ini adalah pengujian performa yang berfokus pada analisis teknis pada *client side*.

Lighthouse mengukur kinerja halaman dengan mensimulasikan interaksi pengguna dan menganalisis serangkaian metrik atau parameter yang dikenal sebagai *Web Vitals*, meliputi *First Contentful Paint* (FCP) untuk mengukur waktu *render* halaman pertama, *Largest Contentful Paint* (LCP) untuk mengetahui kapan konten utama tampil sepenuhnya, serta *Cumulative Layout Shift* (CLS) untuk menilai stabilitas visual selama proses pemuatan. Hasil kalkulasi dari metrik-metrik tersebut kemudian dikonversi menjadi skor performa keseluruhan dengan skala 0 hingga 100 [20]. Berdasarkan standar penilaian *Lighthouse*, skor performa dikategorikan menjadi tiga tingkat, yaitu 90 hingga 100 termasuk dalam kategori baik (*good*). Skor 50 hingga 89 termasuk dalam kategori perlu perbaikan (*needs improvement*) dan skor 0 hingga 49 masuk dalam kategori buruk (*poor*).

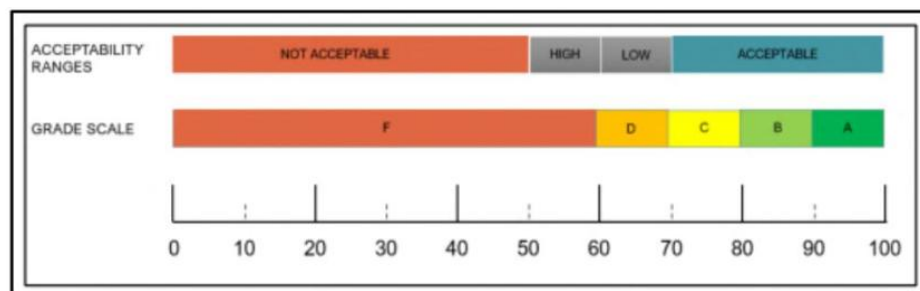
2.15 System Usability Scale (SUS)

System Usability Scale atau SUS merupakan metode pengujian *usability* suatu sistem secara sederhana dengan menggunakan sepuluh pertanyaan yang memberikan pandangan menyeluruh terhadap tingkat kebermanfaatan suatu sistem.

SUS menggunakan skala *Likert* yang mengharuskan responden memberikan tingkat persetujuan atau ketidaksetujuan terhadap setiap pertanyaan dengan skala 5 poin.

Instrumen SUS terdiri atas 10 pernyataan yang dijawab oleh partisipan berdasarkan tingkat persetujuan terhadap suatu sistem yang sedang diuji. Setiap pertanyaan memiliki jawaban dalam skala 1 sampai 5, nilai 1 berarti sangat tidak setuju dan nilai 5 berarti sangat setuju. Untuk menghitung skor SUS, setiap pertanyaan diberi bobot nilai antara 0 sampai dengan 4. Pada pernyataan nomor ganjil, nilai jawaban dikurangi 1, sedangkan untuk pernyataan nomor genap, nilai 5 dikurangi dengan nilai jawaban responden. Jumlah total skor kemudian dikalikan 2,5 sehingga menghasilkan nilai akhir dengan rentang 0 – 100 [21].

Hasil skor SUS selanjutnya diinterpretasikan dengan menggunakan kategori *adjective rating*, *grade scale* dan *acceptability*. Pada kategori *adjective rating*, skala 0 sampai 25 termasuk dalam kategori *worst imaginable*, skala 25,1 sampai dengan 51,60 masuk dalam kategori *poor*, skala 51,70 sampai dengan 71,00 masuk dalam kategori OK atau *fair*, skala 71,10 sampai dengan 80,70 masuk dalam kategori *good*, skala 80,80 sampai dengan 84,00 masuk dalam kategori *excellent* dan skala 84,10 sampai 100 masuk ke dalam kategori *best imaginable* [22]. Sedangkan untuk menentukan *grade scale* skala yang digunakan adalah 0 sampai 60 untuk *grade scale* F, 61 sampai 70 untuk *grade scale* D, 71 sampai dengan 80 untuk *grade scale* C, 81 sampai dengan 90 untuk *grade scale* B dan 91 sampai dengan 100 untuk *grade scale* A. Lalu untuk *acceptability ranges* menggunakan skala 0 sampai 60 untuk *not acceptable* dan 61 sampai 100 untuk *acceptable* [21].



Gambar 5. Kategori Penilaian SUS

Penentuan jumlah responden dalam pengujian kegunaan sistem sangat berpengaruh terhadap tingkat validitas temuan masalah. Dalam pengujian *usability*, keterlibatan 5 orang responden hanya mampu mengidentifikasi minimal 55% dari total permasalahan yang ada pada sistem. Untuk meningkatkan akurasi hasil, penambahan jumlah sampel menjadi 10 responden dapat meningkatkan persentase temuan masalah menjadi minimal 80%. Namun, untuk mencapai tingkat kepercayaan yang lebih tinggi dalam mendeteksi potensi permasalahan antarmuka, diperlukan setidaknya 15 responden agar dapat menemukan minimal 90% permasalahan, serta 20 responden untuk mendapatkan 95% permasalahan pada sistem [23].

2.16 Penelitian Terdahulu

Penelitian mengenai integrasi model kecerdasan buatan dengan antarmuka web telah banyak dilakukan untuk meningkatkan kemudahan akses dan interaktivitas pengguna. Pentingnya arsitektur modern yang memisahkan *frontend* dan *backend* dibahas dalam penelitian oleh M. Zeiler dkk, melalui pengembangan CoTranslate, yaitu sebuah platform berbasis web untuk mengumpulkan dan mengolah data secara masal (*crowdsourcing*). Dalam penelitiannya, ia menegaskan bahwa pemisahan arsitektur yang jelas memungkinkan sistem untuk menangani interaksi banyak pengguna sekaligus saat mengirimkan data ke sistem secara efisien [24].

Mengikuti prinsip arsitektur tersebut, banyak penelitian telah membuktikan juga mengenai keberhasilan fungsional dari integrasi model kecerdasan buatan ke dalam web. Salah satu contohnya adalah penelitian oleh Fauzan Nuraulia Darmawan dkk, yang berjudul “*Sistem Klasifikasi Penyakit Kulit Menggunakan Metode Convolutional Neural Network (CNN) Berbasis Website*”. Pada penelitian ini dikembangkan sistem berbasis web yang dapat melakukan klasifikasi penyakit kulit seperti Dermatitis Atopik, Dermatitis Numular dan Dermatitis Kontak secara otomatis menggunakan model CNN. Sistem tersebut dirancang memanfaatkan *framework* Flask pada sisi *backend* serta menggunakan HTML, CSS dan JavaScript

dalam pembangunan antarmuka pengguna. Hasil penelitian menunjukkan bahwa integrasi model CNN ke dalam platform web dapat berjalan dengan baik, sistem mampu menampilkan hasil klasifikasi melalui antarmuka secara langsung dengan tingkat akurasi mencapai 87% [25]. Pada penelitian ini terlihat bahwa integrasi antara model CNN dengan antarmuka berbasis web berpotensi besar untuk mempermudah proses identifikasi dan visualisasi hasil prediksi kepada pengguna.

Tren serupa juga ditunjukkan oleh Evi Marlina dan Nursyah Fanni dalam penelitiannya yang berjudul “*Pembuatan Sistem Absensi Praktikum dengan Pengenalan Wajah Menggunakan Metode CNN Berbasis Web*” juga menerapkan integrasi model CNN ke dalam sistem berbasis web. Penelitian ini bertujuan untuk menggantikan metode absensi manual dengan sistem otomatis berbasis pengenalan wajah. Sistem dibangun menggunakan bahasa Python dan *framework* Flask pada sisi *backend*, serta HTML CSS dan JavaScript untuk antarmuka web [26]. CNN digunakan untuk mengenali identitas mahasiswa melalui citra wajah dengan tingkat akurasi tinggi dan memungkinkan pencatatan kehadiran secara otomatis dan *real time*. Pengujian dilakukan dengan metode *black box testing* dan hasilnya menunjukkan bahwa seluruh fitur berjalan dengan baik. Namun penelitian ini belum menitikberatkan pada metode pengembangan yang sistematis serta belum mengoptimalkan aspek pengguna, karena fokus utamanya pada fungsionalitas dan kinerja model CNN.

Lebih lanjut, pada penelitian oleh Wulan Adira Nasution dkk, dengan judul “*Implementasi Metode Convolutional Neural Network (CNN) dalam Klasifikasi Tanaman Herbal Berbentuk Rimpang pada Sistem Informasi Berbasis Website*” mengembangkan sistem berbasis web yang diintegrasikan dengan model CNN untuk melakukan klasifikasi tanaman herbal seperti jahe, kunyit dan lengkuas. Sistem ini memungkinkan pengguna mengunggah citra tanaman melalui halaman web dan menampilkan klasifikasi dalam bentuk label prediksi beserta tingkat kemiripannya. Model CNN yang digunakan adalah Xception, dengan tingkat akurasi mencapai 92,73% serta nilai rata-rata *precision* dan *recall* sebesar 0,93 [27]. Antarmuka dibangun dengan HTML, CSS dan JavaScript dengan beberapa

halaman seperti *Home*, *Product*, *Predict* dan *About*, yang dirancang agar informatif dan mudah digunakan.

Di balik keberhasilan fungsional dari model kecerdasan buatan, kerap muncul isu lanjutan yang tidak kalah penting, yaitu aksesibilitas dan pengalaman pengguna. Penelitian yang dilakukan oleh Enisda Libra Kelana dkk, yang berjudul “*Integrating the CNN Model with the Web for Indonesian Sign Language (BISINDO) Recognition*” menyoroti masalah utama yaitu model-model CNN yang berpotensi tinggi sering kali memiliki aksesibilitas yang terbatas. Dalam penelitian tersebut dihasilkan model CNN untuk mengenali abjad bahasa isyarat Indonesia (BISINDO) dan mengintegrasikannya ke dalam platform web menggunakan Python dengan *library* gradio. Penelitian ini membuktikan bahwa model AI yang akurat tetap sangat memerlukan antarmuka web yang praktis agar dapat diakses oleh pengguna [28]. Namun, dari sisi tampilan web tersebut hanya berfokus pada pembuatan antarmuka yang cepat, sehingga memiliki celah untuk pengembangan *frontend* yang lebih terintegrasi.

Relevansi integrasi model kecerdasan buatan ke dalam platform web juga ditunjukkan oleh Mutiara Vebriani dan Witi Yustanti dalam penelitian yang berjudul “*Klasifikasi Deteksi Link Phising DANA Kaget Menggunakan Metode Support Vector Machine Berbasis Website*”. Dalam penelitian tersebut, berhasil dikembangkan sebuah sistem berbasis web yang mampu mendeteksi tautan *phising* dengan akurasi tinggi menggunakan model SVM. Sistem ini berhasil menyediakan antarmuka sederhana, pengguna dapat memasukkan tautan dan menerima hasil prediksi secara langsung. Penelitian ini membuktikan bahwa nilai guna sebuah model AI dapat dimaksimalkan ketika diimplementasikan dalam sebuah web yang fungsional dan mudah diakses [29]. Namun, implementasi *frontend* pada penelitian tersebut masih menggunakan pendekatan dasar, sehingga membuka peluang untuk penerapan metode pengembangan yang lebih terstruktur seperti RAD dan pemanfaatan *framework*.

Pendekatan untuk mengatasi keterbatasan *frontend* tidak hanya bersifat teknis, tetapi juga metodologis. Seperti pada penelitian oleh Michael Zeller dkk, yang mengembangkan aplikasi bernama “digiDEM-SCREEN” dengan menerapkan pendekatan UCD secara iteratif, prototipe aplikasi diuji langsung kepada pengguna target dalam beberapa tahap. Untuk mengukur tingkat kemudahan penggunaan, penelitian tersebut secara kuantitatif menggunakan metode SUS. Proses iteratif yang berfokus pada *feedback* pengguna terbukti berhasil meningkatkan kualitas antarmuka secara signifikan, yang ditunjukkan dengan peningkatan skor SUS dari 72,5 pada versi awal menjadi 82,4 pada versi berikutnya[30].

Metodologi yang berpusat pada pengguna ini perlu didukung oleh teknologi *frontend* yang tepat. Pada penelitian yang dilakukan oleh Hamed Hussen Ben Kora dan Mohamed Sasi Manita, diidentifikasi bahwa React sebagai *library* unggul dalam *client side rendering* (CSR), namun menyoroti kelemahannya mengenai Search Engine Optimization (SEO) dan kebutuhan akan pustaka eksternal untuk *routing*. Sebagai framework yang dibangun diatas React, Next.js disajikan sebagai solusi untuk mengatasi keterbatasan tersebut melalui implementasi *server side rendering* (SSR) untuk meningkatkan kinerja SEO dan menyediakan mekanisme *routing* yang sudah terintegrasi [31].

Keunggulan teoritis Next.js ini telah dibuktikan pada jurnal berjudul “*AgriScan: Next.js powered cross-platform solution for automated plant disease diagnosis and crop health management*” yang berfokus pada pengembangan sistem lintas platform untuk diagnosis penyakit tanaman secara otomatis menggunakan model CNN. Penelitian ini menggunakan Next.js sebagai *framework* utama untuk membangun antarmuka *frontend* yang responsif dan interaktif. Dari sistem yang dikembangkan, dapat memungkinkan petani untuk mengunggah gambar daun tanaman dan menerima hasil klasifikasi penyakit pada tanaman secara *realtime* dengan akurasi model mencapai 93,45%. Studi ini menunjukkan keberhasilan implementasi Next.js dalam menjembatani model *deep learning* yang kompleks dengan pengguna, serta memperkuat argumen bahwa Next.js merupakan pilihan

yang kuat untuk membangun aplikasi berbasis AI yang efisien dan mudah diakses di berbagai perangkat[32].

Keberhasilan implementasi Next.js juga diperkuat oleh penelitian dengan judul “*Hemaclassify: Web-based Blood Diseases Classification System Utilizing High-Resolution Neural Networks*” dikembangkan sebuah sistem pakar berbasis web untuk klasifikasi penyakit darah secara otomatis menggunakan model *High-Resolution Neural Networks* (HRNet). Dalam penelitian tersebut juga digunakan teknologi *frontend* seperti Next.js, Tailwind CSS dan komponen Shadcn UI untuk membangun antarmuka pengguna. Penelitian ini membuktikan bahwa kombinasi teknologi tersebut mampu menghasilkan aplikasi yang tidak hanya akurat dari sisi model AI, tetapi juga memiliki tingkat penerimaan dan kegunaan yang tinggi, seperti yang ditunjukkan oleh hasil *User Acceptance Testing* (UAT) yang memperoleh skor kepuasan untuk fungsionalitas dan kegunaan mencapai 4/5 hingga 5/5[33].

Berdasarkan beberapa penelitian terdahulu tersebut, dapat disimpulkan bahwa integrasi antara model kecerdasan buatan yang salah satunya Metode *Convolutional Neural Network* (CNN) dengan sistem berbasis web telah memberikan kontribusi besar dalam pengembangan aplikasi kecerdasan buatan yang mudah diakses. Meskipun beberapa penelitian telah menunjukkan pentingnya penerapan metodologi pengembangan sistematis seperti RAD dan UCD, namun sebagian besar penelitian lainnya masih lebih berfokus pada aspek fungsionalitas dan akurasi model. Akibatnya aspek *frontend* seperti desain antarmuka dan pengembangan aplikasinya belum menjadi prioritas utama. Selain itu, pemanfaatan teknologi modern seperti *library* React.js dan *framework* Next.js dalam pengembangan antarmuka sistem berbasis kecerdasan buatan masih terbatas, sehingga menyisakan ruang untuk inovasi.

Oleh karena itu, penelitian ini dilakukan untuk mengisi celah tersebut dengan menerapkan metode RAD terkhusus dalam pengembangan *frontend website* sistem

klasifikasi spesies lebah madu Indo-Malaya berbasis model CNN, menggunakan *library* React dan *framework* Next.js.

III. METODE PENELITIAN

3.1 Waktu dan Tempat Penelitian

Penelitian ini dilaksanakan pada November 2025 sampai dengan Februari 2026, penelitian ini bertempat di Suhita Bee Farm, Lembah Suhita, Kec. Teluk Betung Utara, Kota Bandarlampung. Berikut adalah tabel jadwal kegiatan penelitian yang dilakukan.

Tabel 5 Jadwal Kegiatan Penelitian

No	Aktivitas	November 2025		Desember 2025				Januari 2026				Februari 2026			
		3	4	1	2	3	4	1	2	3	4	1	2	3	4
1.	Perencanaan Kebutuhan														
2.	Desain Sistem														
3.	Pengembangan														
4.	Implementasi														
5.	Penulisan Laporan														

3.2 Alat dan Bahan Penelitian

Dalam penelitian ini, digunakan beberapa alat bantu dan bahan utama yang menunjang proses pengembangan *frontend website* sistem klasifikasi lebah madu. Alat penelitian terbagi menjadi perangkat keras dan perangkat lunak, serta bahan

penelitian berfokus pada sumber daya data yang diakses dari *backend* melalui API dan data statis yang akan ditampilkan pada antarmuka.

Tabel 6 Alat Penelitian (*Hardware*)

No.	Nama Perangkat Keras	Spesifikasi	Fungsi
1.	Laptop	Prosesor AMD Ryzen 5 5600H, RAM 16 GB, SSD 512 GB, GPU NVIDIA GeForce GTX 1650, Layar 15,6 inci FHD	Digunakan untuk melakukan aktivitas selama proses penelitian berlangsung, seperti perancangan, desain sistem, pengembangan dan implementasi.

Tabel 7 Alat Penelitian (*Software*)

No.	Nama Perangkat Lunak	Fungsi
1.	Windows 11	Sistem operasi utama yang digunakan untuk menjalankan seluruh perangkat lunak pengembangan.
2.	Draw IO	Digunakan untuk membuat diagram perancangan seperti <i>use case</i> dan <i>activity</i>
3.	Figma	Digunakan untuk merancang <i>low-fi</i> dan <i>high-fi</i> tampilan antarmuka <i>website</i> sebelum dikembangkan.
4.	Visual Studio Code	<i>Code editor</i> yang digunakan untuk menulis dan mengelola kode proyek.
5.	React.js	<i>Library</i> JavaScript yang digunakan untuk membangun antarmuka yang interaktif.
6.	Next.js	<i>Framework</i> berbasis React yang digunakan untuk mengatur struktur proyek, <i>routing</i> , serta mendukung <i>server-side rendering</i> (SSR).
7.	Node.js	<i>Runtime environment</i> JavaScript yang digunakan untuk menjalankan server pengembangan lokal, menginstal depedensi, serta melakukan proses <i>build</i> sistem sebelum di- <i>deploy</i> .
8.	Shaden UI	Kumpulan komponen antarmuka siap pakai berbasis Tailwind CSS untuk mempercepat proses pengembangan UI.
9.	Git dan Github	Untuk mengelola <i>version control</i> dan menyimpan repositori secara daring.
10.	Vercel	Digunakan untuk <i>deployment website</i> agar dapat diakses secara publik melalui browser.
11.	Google Chrome	Browser utama yang digunakan untuk menjalankan dan menguji tampilan <i>website</i> , termasuk responsivitas di beberapa ukuran layar.
12.	Postman	Digunakan untuk menguji koneksi RESTful API ketika akan diintegrasikan.

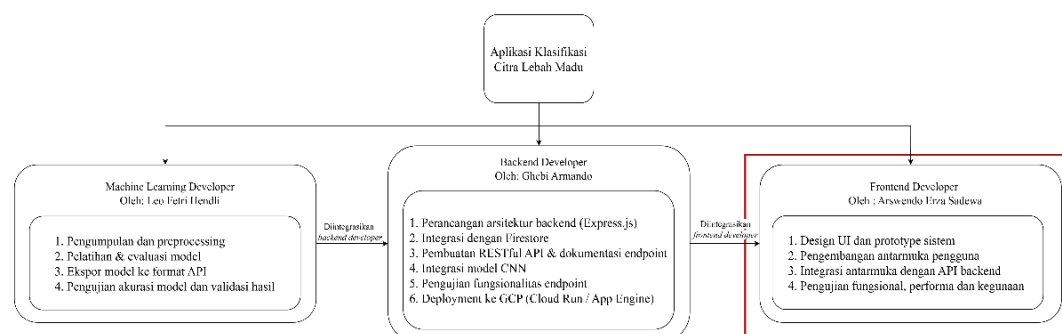
No.	Nama Perangkat Lunak	Fungsi
13.	Microsoft Excel	Digunakan untuk mengolah dan menganalisis data hasil pengujian fungsionalitas dan <i>usability</i> .
14.	Google Lighthouse	Digunakan untuk menguji performa dan aksesibilitas dari <i>frontend website</i>

Tabel 8 Bahan Penelitian

No.	Jenis Bahan Penelitian	Fungsi
1.	Data Hasil Klasifikasi Lebah	Sebagai <i>input</i> data utama pasca pengunggahan gambar untuk menampilkan hasil klasifikasi
2.	Data Informasi Spesies	Data pendukung yang mencakup informasi umum mengenai spesies lebah madu (deskripsi, karakteristik, dll) untuk ditampilkan pada bagian informasi atau detail hasil klasifikasi
3.	<i>Endpoint</i> RESTful API	Digunakan untuk integrasi koneksi antara antarmuka dengan <i>backend</i>
4.	Aset Visual dan Media	Digunakan sebagai bahan visual dalam perancangan antarmuka dan implementasi <i>frontend</i> untuk memperkaya tampilan

3.3 Tahapan Penelitian

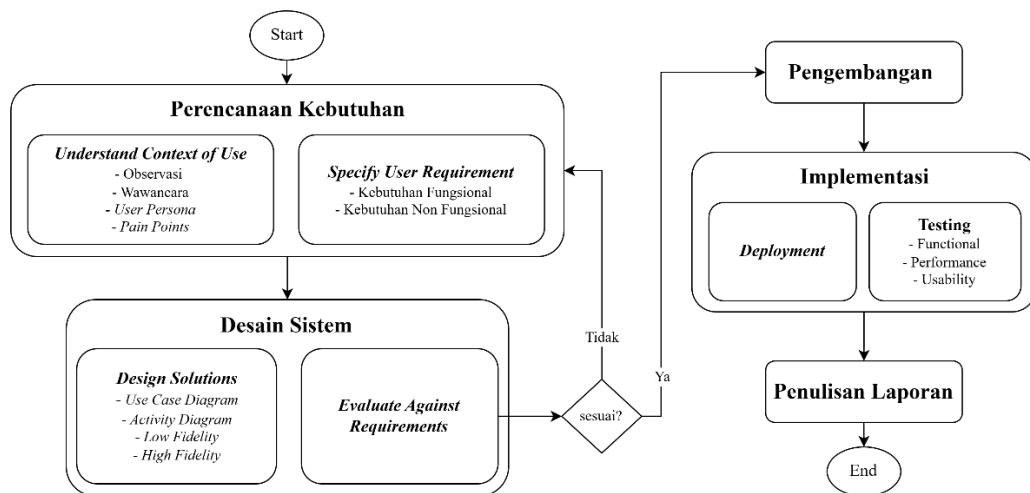
Perancangan sistem klasifikasi citra spesies lebah madu ini, dibagi ke dalam 3 anggota tim penelitian dengan fokus dan pembagian tugas masing-masing. Tim tersebut terdiri dari *Machine Learning Developer*, *Backend Developer*, dan *Frontend Developer* dengan pembagian tugas seperti pada gambar dibawah ini.



Gambar 6. Pembagian Pekerjaan Tim Penelitian

Metode penelitian yang digunakan dalam pengembangan *frontend website* sistem klasifikasi lebah madu Indo-Malaya ini mengacu pada pendekatan RAD yang diintegrasikan dengan prinsip-prinsip UCD. Tahapan penelitian ini terdiri dari 5 fase utama, yaitu Perencanaan Kebutuhan, Design Sistem, Pengembangan, Implementasi, serta Penulisan Laporan sebagai tahapan paling akhir. Tahapan penelitian diawali dengan Perencanaan Kebutuhan yang dibagi menjadi dua bagian utama, yaitu *Understand Context of Use* yang meliputi observasi dan wawancara sehingga menghasilkan *user persona* dan *pain points*, serta *Specify User Requirement* untuk merumuskan kebutuhan fungsional dan nonfungsional sistem. Tahap selanjutnya adalah Desain Sistem yang mencakup *Design Solutions* melalui penyusunan *use case diagram*, *activity diagram*, serta perancangan *Low Fidelity* dan *high fidelity*. Setelah rancangan visual selesai dilakukan tahapan *Evaluate Against Requirement* untuk memvalidasi desain kepada *stakeholder*. Pada tahap ini terdapat proses iteratif, yaitu ketika hasil evaluasi rancangan belum sesuai dengan kebutuhan pengguna, maka proses akan kembali melakukan perbaikan melalui fase perencanaan kebutuhan kembali.

Setelah rancangan dinyatakan sesuai, proses dilanjutkan pada tahapan Pengembangan, hasil desain diimplementasikan ke dalam baris kode program (*coding*). Tahap berikutnya adalah Implementasi yang terdiri dari proses *Deployment* sistem ke server agar dapat diakses oleh publik serta tahap *Testing* yang mencakup pengujian fungsional (*black box testing*), pengujian performa dengan Google Lighthouse dan pengujian kegunaan dengan metode SUS. Tahapan terakhir adalah Penulisan Laporan yang mendokumentasikan seluruh rangkaian proses penelitian dan hasil pengembangan sistem yang telah dilakukan.



Gambar 7. Tahapan Penelitian

3.3.1 Perencanaan Kebutuhan

Tahapan penelitian diawali dengan perencanaan kebutuhan yang terintegrasi dengan prinsip dari UCD dan terbagi menjadi dua fokus, yaitu *Understand Context of Use* dan *Specify User Requirement*. Pada fase *Understand Context of Use* dilakukan observasi dan wawancara dengan *stakeholder*, yaitu pengelola Suhita Bee Farm sebagai praktisi lebah yang mengetahui dan memahami kendala dalam informasi dan identifikasi spesies lebah di lapangan. Proses ini bertujuan untuk menghasilkan *User Persona* dan pemetaan *pain points* pengguna. Tujuannya adalah untuk memperoleh pemahaman mengenai kebutuhan, permasalahan dan ekspektasi pengguna terhadap antarmuka sistem yang dikembangkan. Selanjutnya *Specify User Requirement* dilakukan dengan menganalisis hasil dan pemahaman yang telah diperoleh untuk menentukan fitur-fitur sistem, serta dihasilkan kebutuhan fungsional dan nonfungsional yang akan menjadi acuan pada proses perancangan.

Tabel 9 Tahapan Perencanaan Kebutuhan

<i>Input</i>	<i>Proses</i>	<i>Output</i>
Pengelola <i>Suhita Bee Farm</i>	Wawancara dan observasi langsung	<i>User persona</i> dan <i>pain points</i>

<i>Input</i>	<i>Proses</i>	<i>Output</i>
Hasil UCD (<i>user persona</i> dan <i>pain points</i>)	Analisis kebutuhan dan penetapan ekspektasi sistem	Kebutuhan fungsional dan nonfungsional sistem

3.3.2 Desain Sistem

Pada tahap ini, kembali dibagi menjadi 2 bagian yang terintegrasi dengan UCD, yaitu *design solutions* dan *evaluate against requirements*. Dalam sub-tahapan *design solutions* dibuatlah diagram UML yang terdiri dari *use case diagram* dan *activity diagram* yang menggambarkan interaksi antara pengguna dengan sistem. Selanjutnya dibuatlah perancangan visual yang dimulai dengan pembuatan *Low Fidelity* sebagai kerangka dasar yang kemudian dikembangkan menjadi *high fidelity* dengan detail warna, tipografi dan komponen antarmuka.

Setelah seluruh desain selesai maka akan dilakukan sub-tahapan *evaluate against requirements* dengan melakukan pengujian secara langsung kepada *stakeholder* untuk divalidasi dan memberikan *feedback* secara langsung mengenai rancangan yang telah dibuat. Tahap ini bersifat iteratif, yang apabila hasil evaluasi dari *stakeholder* menyatakan bahwa rancangan belum sesuai dengan kebutuhan pengguna, maka akan kembali dilakukan perbaikan melalui proses-proses sebelumnya hingga dinyatakan sesuai. Dengan adanya desain dan rancangan yang sudah terarah, maka proses pengembangan *frontend* ditahap berikutnya dapat dilakukan dengan lebih efisien karena setiap alur dan komponen telah digambarkan dengan jelas.

Tabel 10 Tahapan Desain Sistem

<i>Input</i>	<i>Proses</i>	<i>Output</i>
Kebutuhan fungsional dan non-fungsional	Analisis kebutuhan dan perumusan alur interaksi sistem	<i>Use case diagram</i> dan <i>activity diagram</i> (diagram UML)
Kebutuhan fungsional dan non-fungsional, serta diagram UML.	Perancangan <i>wireframe</i> untuk tata letak dan struktur	<i>Wireframe</i> atau representasi awal tata letak yang disetujui

<i>Input</i>	<i>Proses</i>	<i>Output</i>
<i>Wireframe</i>	Pengembangan desain visual dan gaya	<i>Mockup high-fidelity</i> atau rancangan akhir.
Rancangan Antarmuka	Evaluasi dengan <i>stakeholder</i>	Hasil validasi kesesuaian desain (Iterasi)

3.3.3 Pengembangan

Setelah rancangan desain divalidasi dan dinyatakan sudah sesuai, maka dilakukan tahapan pengembangan. Rancangan antarmuka dan komponen yang telah dibuat pada tahap sebelumnya diimplementasikan menggunakan *framework* Next.js berbasis React.js. Setiap komponen dikembangkan secara modular agar mudah diatur dan dapat digunakan kembali pada bagian halaman lain. Selanjutnya dilakukan integrasi untuk menghubungkan *frontend* terhadap RESTful API yang telah dibuat oleh pengembang *backend*, agar *frontend* dapat menampilkan data informasi dan hasil klasifikasi citra dari model CNN secara langsung, serta informasi mengenai lebah pada halaman web. Pengembangan ini merupakan proses pengubahan desain visual menjadi kode program fungsional sesuai dengan yang telah dirancang.

Tabel 11 Tahapan Pengembangan

<i>Input</i>	<i>Proses</i>	<i>Output</i>
Desain dan diagram UML	Implementasi komponen <i>frontend</i> menggunakan Next.js yang berbasis React.js dan Shadcn UI.	Komponen antarmuka <i>website</i>
Komponen antarmuka dan API sistem dari tim <i>backend</i>	Integrasi api dengan <i>frontend website</i>	Prototipe <i>website</i> yang siap untuk diuji

3.3.4 Implementasi

Sebelum di publikasikan, sistem dijalankan pada *localhost* untuk kembali memastikan seluruh komponen berjalan sesuai dengan rancangan. Pada tahap ini

dilakukan pengujian fungsionalitas dengan metode *black box testing*. Pengujian ini mencakup semua fungsi yang tersedia untuk setiap peran pengguna, yaitu Pengguna Umum, Praktisi dan Super Admin untuk memeriksa setiap alur kerja fungsi, koneksi API, serta ketepatan hasil klasifikasi.

Setelah sistem dinyatakan berjalan dengan baik dan stabil, maka proses *deployment* dilakukan dengan menggunakan *platform* Vercel sehingga sistem dapat diakses secara publik melalui *browser*. Pada tahap ini dilakukan pengujian performa menggunakan *tools* Google Lighthouse untuk mengukur kualitas teknis *frontend* berdasarkan performanya.

Setelah performa sistem dipastikan optimal, tahap selanjutnya dilakukan pengujian *usability* dengan melibatkan pengguna langsung menggunakan metode SUS. Pengujian *usability* difokuskan pada 15 pengguna yang berinteraksi langsung dengan fitur utama sistem, yaitu Pengguna Umum yang mewakili pengguna awam seperti masyarakat atau mahasiswa yang tidak memiliki keahlian khusus di bidang yang terkait dengan lebah, serta Praktisi yang mewakili pengguna dengan pengetahuan dan kepentingan khusus.

Tabel 12 Tahapan Implementasi

<i>Input</i>	<i>Proses</i>	<i>Output</i>
Prototipe <i>website</i>	Pengujian fungsionalitas dengan <i>black box testing</i>	Hasil <i>black box testing</i>
Prototipe yang telah diuji fungsionalitas	<i>Deployment</i> sistem ke publik menggunakan vercel	Sistem <i>frontend</i> klasifikasi citra lebah madu yang dapat diakses publik
<i>Website</i> yang telah di- <i>deploy</i> dan pengguna	Pengujian performa dengan Google Lighthouse dan pengujian <i>usability</i> dengan melibatkan pengguna menggunakan metode SUS	Skor performa dan rekomendasi optimasi, serta, skor SUS dan hasil evaluasi <i>usability</i>

3.3.5 Penulisan Laporan

Tahap penulisan laporan merupakan tahapan akhir dalam penelitian yang berfokus pada penyusunan dokumentasi hasil dari seluruh proses pengembangan yang telah dilakukan. Pada tahap ini dilakukan pengumpulan, analisis dan penyusunan hasil dari setiap fase RAD, mulai dari perencanaan kebutuhan, desain sistem, pengembangan, hingga implementasi dan pengujian sistem. Laporan akhir disusun dalam bentuk tugas akhir atau skripsi yang memuat penjelasan metode, hasil implementasi, serta evaluasi kinerja sistem.

V. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil penelitian dan pengembangan yang dilakukan pada sistem BeeVra, berikut adalah kesimpulan yang disusun untuk menjawab rumusan masalah penelitian:

1. Penerapan metode *Rapid Application Development* (RAD) dalam pengembangan *frontend website* BeeVra terbukti efektif melalui penyelesaian seluruh siklus pengembangan dalam durasi yang cepat yaitu 3 bulan 2 minggu. Keterlibatan pihak Suhita Bee Farm sebagai pengguna sejak tahap awal hingga validasi desain *high-fidelity* juga menjadi faktor efektifitas dalam memastikan fitur yang dikembangkan benar-benar menjawab kebutuhan secara tepat sasaran, sementara pemanfaatan komponen yang dapat digunakan kembali (*software reuse*) melalui *framework* Next.js secara signifikan meminimalkan risiko pengerjaan ulang (*rework*) sekaligus mempercepat transisi dari perancangan ke tahap konstruksi kode program yang modular dan terstruktur.
2. Prinsip-prinsip metode *User Centered Design* (UCD) berhasil diimplementasikan melalui perancangan antarmuka yang didasarkan pada data asli pengguna dan sesuai konteks pada lapangan yang dibuktikan dengan pemetaan *user persona* dan *pain points* di tahap awal sebagai acuan penentuan fitur. Selain itu, keberhasilan ini didukung oleh keterlibatan aktif pengguna di seluruh tahapan penelitian dengan menempatkan pengguna sebagai pusat dari seluruh proses perancangan sistem. Penerapan proses desain yang iteratif juga terbukti efektif dalam menyempurnakan antarmuka melalui siklus perbaikan

yang didasarkan pada umpan balik, sehingga meminimalkan celah antara rancangan sistem dengan kebutuhan pengguna.

3. Pengujian fungsionalitas menggunakan metode *Black Box Testing* menunjukkan hasil 100% valid pada seluruh skenario fitur utama yang dikembangkan. Pengujian performa menggunakan Google Lighthouse memperoleh skor di antara 93 hingga 99 yang menandakan bahwa setiap bagian halaman sudah teroptimasi dengan baik. Sementara hasil pengujian kegunaan (*usability*) dengan menggunakan metode SUS menghasilkan skor rata-rata 83,83 yang menempatkan sistem pada kategori *Acceptable* dengan *Grade B* dan predikat *Excellent* yang membuktikan tingkat kepuasan pengguna yang cukup tinggi terhadap sistem. Dengan demikian, hasil pengujian menunjukkan bahwa *frontend website* BeeVra telah memenuhi standar kualitas fungsional, performa, dan kegunaan secara optimal.

5.2 Saran

Berdasarkan hasil penelitian dan pengembangan yang dilakukan pada sistem BeeVra, terdapat saran pengembangan *frontend website* BeeVra di masa mendatang:

1. Mengembangkan sistem BeeVra menjadi *Progressive Web Apps* (PWA) dengan menerapkan strategi *Offline-First* melalui pemanfaatan *Service Workers* dan *Manifest file*, dengan tujuan untuk menjamin ketersediaan akses data pada kondisi atau area dengan stabilitas jaringan yang rendah, sehingga informasi lebih tetap dapat diakses oleh pengguna tanpa bergantung sepenuhnya pada stabilitas jaringan internet.
2. Melakukan optimasi teknis pada proses pemuatan halaman dengan menambahkan strategi *Preloading* pada aset CSS dan *Preconnect* pada alamat *server* yang terpisah dari platform Vercel dikarenakan ditemukan adanya kendala antrean pemanggilan file CSS (*chaining critical requests*) yang menyebabkan penundaan pemuatan konten dan membuat nilai *Speed Index* (SI) dan *Largest Contentful Paint* (LCP) pada *website* BeeVra belum maksimal.

3. Menambahkan fitur pengecilan ukuran gambar otomatis (*image compression*) pada sisi *frontend* sebelum foto dikirim ke server melalui API, dikarenakan pengunggahan foto lebah dengan resolusi tinggi membutuhkan waktu yang cukup lama dan sangat bergantung pada kecepatan internet pengguna. Dengan menambahkan fitur kompresi di bagian *frontend*, ukuran *file* yang dikirim akan menjadi lebih ringan sehingga mempercepat respon dan menghemat penggunaan kuota data.

DAFTAR PUSTAKA

- [1] E. Safitri dan H. Purnobasuki, *Aplikasi Madu sebagai Aktivator STEM CELL*. Airlangga University Press, 2022. [Daring]. Tersedia pada: <https://s.id/apkstemcell>
- [2] D. De Nart, C. Costa, G. Di Prisco, dan E. Carpana, "Image recognition using convolutional neural networks for classification of honey bee subspecies," *Apidologie*, vol. 53, no. 1, 2022, doi: 10.1007/s13592-022-00918-5.
- [3] M. A. Rojabi, *Pengantar User Interface (UI)*. Afdan Rojabi, 2025. [Daring]. Tersedia pada: <https://s.id/PengantarUI>
- [4] R. F. Aziza, A. Nurmasani, dan M. Azizah, *Teori dan Praktik Desain UI/UX: Studi Kasus Implementasi dengan Metode Design Thinking*. Penerbit Andi, 2024. [Daring]. Tersedia pada: <https://s.id/TeoriPraktikUIUX>
- [5] D. Susanti dan P. Adab, *Desain User Interface (UI)/User Experience (UX) Menggunakan Figma*. Penerbit Adab, 2024. [Daring]. Tersedia pada: <https://s.id/DesainUXUI>
- [6] A. B. Pertiwi dkk., *Desain Komunikasi Visual di Era Revolusi Industri 5.0*. PT. Green Pustaka Indonesia, 2025. [Daring]. Tersedia pada: <https://s.id/dkverarevindustri>
- [7] N. Nirsal dkk., *Buku ajar Analisis dan Perancangan Sistem*. PT. Sonpedia Publishing Indonesia, 2025. [Daring]. Tersedia pada: <https://s.id/analsisisperancangansistem>

- [8] N. Ahmad dkk., *Analisa & Perancangan Sistem Informasi Berorientasi Objek*. Penerbit Widina, 2022. [Daring]. Tersedia pada: <https://s.id/analisperancanganSIBO>
- [9] S. Andriyanto, *Pemodelan Perangkat Lunak Behavior Diagram*, vol. 17. Politeknik Manufaktur Negeri Bangka Belitung, 2022.
- [10] A. P. Hadi, *Mengenal Frontend Development*. Yayasan Prima Agus Teknik, 2024.
- [11] D. Rodrrigues, *Learn Next.js: From Fundamentals to Practical Applications*. StudioD21, 2025. [Daring]. Tersedia pada: <https://s.id/learnNEXTJS>
- [12] H. Kurniawan dkk., *Belajar Web Programming: Referensi Pengenalan Dasar Tahapan Belajar Pemrograman Web Untuk Pemula*. PT. Sonpedia Publishing Indonesia, 2023. [Daring]. Tersedia pada: <https://s.id/dasarpemweb>
- [13] T. Hinrichs, *A UI-Enhanced Approach to Generic Web-Based Scheduling*. Uconn Library, 2024. [Daring]. Tersedia pada: <https://s.id/webbasedscheduling>
- [14] S. Wijayanto dkk., *Buku Ajar Analisa perancangan sistem Informasi*. PT. Sonpedia Publishing Indonesia, 2024. [Daring]. Tersedia pada: <https://s.id/analisperancangansi>
- [15] M. A. Sunarti, *Analisa dan Perancangan Sistem Informasi*. PT. Kimhsafi Alung Cipta, 2025. [Daring]. Tersedia pada: <https://s.id/analisanperancanganSI>
- [16] R. M. Ikhsanuddin, L. H. Annisa, dan D. Rusvinasari, *Buku Ajar Rekayasa Perangkat Lunak*. Uwais Inspirasi Indonesia, 2025. [Daring]. Tersedia pada: <https://s.id/bukuajarRPL>
- [17] D. Susilo dkk., *Konsep Dasar Rekayasa Perangkat Lunak*. Cendikia Mulia Mandiri, 2023. [Daring]. Tersedia pada: <https://s.id/dasarRPL>

- [18] E. Anan dan M. S. Indrajaya, *Perancangan Sistem Akuntansi Pribadi Berbasis Android*. Jejak Pustaka, 2022. [Daring]. Tersedia pada: <https://s.id/sistemakuntansipribadi>
- [19] Y. Yudhanto dan J. A. Hutabarat, *Panduan Dasar Teknologi Virtual Reality (VR)*. Indotama, 2021. [Daring]. Tersedia pada: <https://s.id/dasarteknologi>
- [20] J. Winkler, *Mastering Adobe Commerce Frontend: Build optimized, user-centric e-commerce sites with tailored theme design and enhanced interactivity*. Packt Publishing, 2024. [Daring]. Tersedia pada: <https://s.id/adobecommercefrontend>
- [21] H. B. Nugroho, *Pengembangan Aplikasi Kepuasan Pelanggan Berbasis Web*. GUEPEDIA, 2023. [Daring]. Tersedia pada: <https://s.id/apkkepuasanpelanggan>
- [22] A. K. Nagar, D. S. Jat, G. M. Raventós, dan D. K. Mishra, “Intelligent Sustainable Systems: Selected Papers of WorldS4 2021 Volume 2,” *Lecture Notes in Computer Science*, vol. 2, hlm. 834, 2021, doi: <https://doi.org/10.1007/978-981-16-6369-7>.
- [23] L. Faulkner, “Beyond the five-user assumption: Benefits of increased sample sizes in usability testing,” *Behavior Research Methods, Instruments, & Computers*, vol. 35, no. 3, hlm. 379, 2023, doi: 10.1109/DCABES.2011.32.
- [24] A. Carvallo, I. Jorquera, dan C. Aspillaga, “SoftwareX CoTranslate: A web-based tool for crowdsourcing high-quality sentence pair corpora,” *SoftwareX*, vol. 23, 2023, doi: 10.1016/j.softx.2023.101508.
- [25] F. N. Darmawan, E. P. Silmina, dan T. Hardiani, “Sistem Klasifikasi Penyakit Kulit Menggunakan Metode Convolutional Neural Network (CNN) Berbasis Website,” *LPPM Universitas Aisyiyah Yogyakarta*, vol. 2, hlm. 871–881, 2024.
- [26] E. Marlina dan N. Fanni, “Pembuatan Sistem Absensi Praktikum dengan Pengenalan Wajah Menggunakan Metode CNN Berbasis Web,” *Politeknik Manufaktur Negeri Bangka Belitung*, hlm. 167–186, 2024.

- [27] W. A. Nasution, D. Y. Niska, S. I. Al Idrus, Z. Indra, dan I. Taufik, "Implementasi Metode Convolutional Neural Network (CNN) Dalam Klasifikasi Tanaman Herbal Berbentuk Rimpang Pada Sistem Informasi Berbasis Website," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 9, no. 5, 2025, doi: 10.36040/jati.v9i5.15092.
- [28] E. L. Kelana, M. R. Anshori Prasetya, Mambang, dan M. Zulfadhilah, "Integrating the CNN Model with the Web for Indonesian Sign Language (BISINDO) Recognition," *Journal of Applied Informatics and Computing*, vol. 9, no. 3, 2025, doi: 10.30871/jaic.v9i3.9345.
- [29] M. Vebriani dan W. Yustanti, "Klasifikasi Deteksi Link Phising DANA Kaget Menggunakan Metode Support Vector Machine Berbasis Website," *Journal of Informatics and Computer Science (JINACS)*, vol. 6, no. 02, hlm. 408–416, 2024, doi: 10.26740/jinacs.v6n02.p408-416.
- [30] M. Zeiler *dkk.*, "A User-Centered Design Approach for a Screening App for People With Cognitive Impairment (digiDEM-SCREEN): Development and Usability Study," *JMIR Hum. Factors*, vol. 12, 2025, doi: 10.2196/65022.
- [31] H. H. Ben kora dan M. S. Manita, "Modern Front-End Web Architecture Using React.js and Next.js," *University of Zawia Journal of Engineering Sciences and Technology*, vol. 2, no. 1, hlm. 1–13, 2024, doi: 10.26629/uzjest.2024.01.
- [32] T. A. Seyam dan A. Pathak, "AgriScan: Next.js powered cross-platform solution for automated plant disease diagnosis and crop health management," *Journal of Electrical Systems and Information Technology*, vol. 11, no. 1, 2024, doi: 10.1186/s43067-024-00169-7.
- [33] K. Arjunan, A. Adam, Raja Zahratul Azma Raja Sabudin, Mohammad Faizal Ahmad Fauzi, dan Elaine Wan Ling Chan, "Hemaclassify: Web-based Blood Diseases Classification System Utilizing High-Resolution Neural Networks," *Applied Mathematics and Computational Intelligence (AMCI)*, vol. 14, no. 3, hlm. 119–138, 2025, doi: 10.58915/amci.v14i3.1021.