

**PENGEMBANGAN *FRONT-END* SISTEM ASESMEN TALENTA
MAHASISWA UNIVERSITAS LAMPUNG BERBASIS NEXT.JS DENGAN
METODE *RAPID APPLICATION DEVELOPMENT***

(Skripsi)

Oleh

**FARIS ILHAM HIDAYAT
NPM 2215061118**



**FAKULTAS TEKNIK
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2026**

**PENGEMBANGAN *FRONT-END* SISTEM ASESMEN TALENTA
MAHASISWA UNIVERSITAS LAMPUNG BERBASIS NEXT.JS DENGAN
METODE *RAPID APPLICATION DEVELOPMENT***

Oleh

FARIS ILHAM HIDAYAT

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 *FRONT-END* SISTEM ASESMEN TALENTA MAHASISWA UNIVERSITAS LAMPUNG BERBASIS NEXT.JS DENGAN METODE *RAPID APPLICATION DEVELOPMENT*

Oleh

FARIS ILHAM HIDAYAT

Unit Penunjang Akademik Pengembangan Karier dan Kewirausahaan Universitas Lampung mengadaptasi konsep *9-Box Talent Matrix* untuk mendukung pemetaan dan pengembangan potensi mahasiswa. Penelitian ini bertujuan mengembangkan *front-end* sistem Asesmen Talenta Mahasiswa berbasis *web* menggunakan *framework* Next.js yang mampu memvisualisasikan konsep *9-Box Talent Matrix* dengan mengintegrasikan instrumen Minat Karier, Karakteristik Diri, dan Kesejahteraan Psikologis. Pengembangan sistem dilakukan menggunakan metode *Rapid Application Development* melalui 3 (tiga) iterasi dan menghasilkan 11 fungsionalitas berupa autentikasi pengguna, asesmen mahasiswa, visualisasi statistik, manajemen data, log aktivitas, serta pengelolaan profil pengguna. Pengujian sistem dilakukan menggunakan *Black-Box Testing* melalui 156 skenario dengan tingkat keberhasilan 100%, evaluasi performa menggunakan Google Lighthouse, serta pengujian *usability* melalui *System Usability Scale* dan *Eight Golden Rules*. Hasil pengujian menunjukkan performa sistem yang sangat baik dengan skor Google Lighthouse sebesar 99 dan nilai *Largest Contentful Paint* sebesar 1,0 detik. Evaluasi *usability* menghasilkan skor SUS sebesar 75,87 dengan kategori “Baik” (*Good*) dan skor *Eight Golden Rules* sebesar 164,4 dengan kategori “Sangat Baik”. Dengan demikian, sistem yang dikembangkan terbukti fungsional, andal, dan layak digunakan sebagai solusi digital dalam mendukung strategi pengembangan potensi mahasiswa di Universitas Lampung.

Kata kunci: Asesmen Talenta, *Front-End*, *Website*, Next.js, *Rapid Application Development*

ABSTRACT

FRONT-END DEVELOPMENT OF A STUDENT TALENT ASSESSMENT SYSTEM AT THE UNIVERSITY OF LAMPUNG USING NEXT.JS WITH THE RAPID APPLICATION DEVELOPMENT METHOD

By

FARIS ILHAM HIDAYAT

Center for Career and Entrepreneurship Development at the University of Lampung adopts the 9-Box Talent Matrix concept to support the mapping and development of student potential. This study aims to develop the front-end of a web-based Student Talent Assessment system using the Next.js framework that is able to visualize the 9-Box Talent Matrix concept by integrating Career Interest, Self-Characteristics, and Psychological Well-Being instruments. The system development was conducted using the Rapid Application Development method through 3 (three) iterations and produced 11 functionalities including user authentication, student assessment, statistical visualization, data management, activity logs, and user profile management. System testing was carried out using Black-Box Testing through 156 test scenarios with a 100% success rate, performance evaluation using Google Lighthouse, and usability testing using the System Usability Scale and Eight Golden Rules. The testing results showed excellent system performance with a Google Lighthouse score of 99 and a Largest Contentful Paint value of 1.0 seconds. The usability evaluation produced a SUS score of 75.87 categorized as “Good” and an Eight Golden Rules score of 164.4 categorized as “Very Good”. Therefore, the developed system is proven to be functional, reliable, and feasible to be implemented as a digital solution to support student potential development strategies at the University of Lampung.

Keywords: Talent Assessment, Front-End, Website, Next.js, Rapid Application Development

Judul Skripsi : PENGEMBANGAN *FRONT-END* SISTEM
ASESMEN TALENTA MAHASISWA
UNIVERSITAS LAMPUNG BERBASIS
NEXT.JS DENGAN METODE *RAPID*
APPLICATION DEVELOPMENT

Nama Mahasiswa : Faris Ilham Hidayat

Nomor Pokok Mahasiswa : 2215061118

Program Studi : Teknik Informatika

Fakultas : Teknik

MENYETUJUI

1. Komisi Pembimbing

Pembimbing Utama

Pembimbing Pendamping



Dr. Eng. Ir. Mardiana, S.T., M.T., IPM.
NIP.197203161999032002



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

2. Mengetahui

Ketua Jurusan
Teknik Elektro

Ketua Program Studi
Teknik Informatika



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

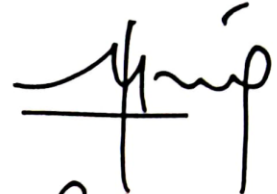


Yessi Mulyani, S.T., M.T.
NIP.197312262000122001

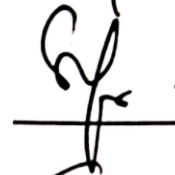
MENGESAHKAN

1. Tim Penguji

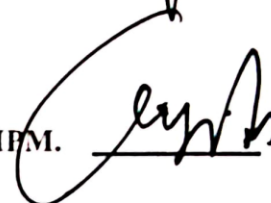
Ketua : Dr. Eng. Ir. Mardiana, S.T., M.T., IPM.



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



Penguji : Ir. Gigih Forda Nama, S.T., M.T.I., IPM.



2. Dekan Fakultas Teknik



Dr. H. Ahmad Herison, S.T., M.T.
NIP. 196910302000031001

Tanggal Lulus Ujian Skripsi : 9 Juni 2026

SURAT PERNYATAAN

Saya yang bertanda tangan di bawah ini :

Nama : Faris Ilham Hidayat

NPM : 2215061118

Dengan ini menyatakan bahwa skripsi yang saya susun dengan judul “Pengembangan *Front-End* Sistem Asesmen Talenta Mahasiswa Universitas Lampung Berbasis Next.Js dengan Metode *Rapid Application Development*” sepenuhnya merupakan hasil dari karya asli saya sendiri. Seluruh isi telah disusun sesuai dengan kaidah penulisan karya ilmiah yang berlaku di Universitas Lampung. Jika di kemudian hari terbukti bahwa skripsi ini merupakan salinan atau dikerjakan oleh pihak lain, maka saya bersedia menerima sanksi sesuai dengan ketentuan hukum maupun peraturan akademik yang berlaku.

Bandar Lampung, 9 Juni 2026



4F:647ANX404993506
Faris Ilham Hidayat
NPM.2215061118

RIYAWAT HIDUP



Penulis bernama lengkap Faris Ilham Hidayat yang lahir di Way Ngison pada tanggal 13 September 2004, anak pertama dari Bapak Suparlan dan Ibu Rohmah Maleha. Penulis menempuh pendidikan dimulai dari Pendidikan Anak Usia Dini (PAUD) Kinasih Desa Way Ngison pada tahun 2008-2010, sekolah dasar di SDN 1 Gemahripah pada tahun 2010-2016, sekolah menengah pertama di SMPN 1 Pagelaran pada tahun 2016-2019, dan sekolah menengah atas di SMAS Muhammadiyah 1 Sendang Agung pada tahun 2019-2022. Pada tahun 2022, penulis diterima sebagai mahasiswa Program Studi Teknik Informatika, Universitas Lampung melalui jalur Seleksi Bersama Masuk Perguruan Tinggi Negeri (SBMPTN). Selama menjadi mahasiswa, penulis pernah mengikuti beberapa kegiatan, di antaranya:

1. Menjadi anggota Divisi Media dan Informasi Forum Komunikasi Bidikmisi/KIP-Kuliah Universitas Lampung pada tahun kepengurusan 2024.
2. Menjadi Ketua Divisi Media dan Informasi Forum Komunikasi Bidikmisi/KIP-Kuliah Universitas Lampung pada tahun kepengurusan 2025.
3. Mengikuti program Wirausaha Merdeka Batch 3 - PROTEKSI UNILA yang diselenggarakan oleh Universitas Lampung pada tahun 2024.

Pada tahun 2026, sebagai syarat untuk mencapai gelar Sarjana Teknik (S.T.) di Universitas Lampung, penulis menyelesaikan tugas akhir skripsi dengan judul “Pengembangan *Front-End* Sistem Asesmen Talenta Mahasiswa Universitas Lampung Berbasis Next.js dengan Metode *Rapid Application Development*”. Melalui gelar dan ilmu yang diperoleh, penulis berharap dapat menjadi pribadi yang senantiasa menebar manfaat dan memberikan kontribusi nyata bagi sesama.

MOTO

فَإِنَّ مَعَ الْعُسْرِ يُسْرًا

“Maka, sesungguhnya beserta kesulitan ada kemudahan.”

(QS. Al-Insyirah: 5)

خَيْرُ النَّاسِ أَنْفَعُهُمْ لِلنَّاسِ

“Sebaik-baik manusia adalah yang paling bermanfaat bagi manusia (lainnya).”

(H.R. Ahmad)

“Jangan takut jatuh, karena yang tidak pernah memanjatlah yang tidak pernah jatuh. Jangan takut gagal, karena yang tidak pernah gagal hanyalah orang-orang yang tidak pernah melangkah. Jangan takut salah, karena dengan kesalahan yang pertama kita dapat menambah pengetahuan untuk mencari jalan yang benar pada langkah yang kedua.”

(Buya Hamka)

“Pada akhirnya ini semua. Hanyalah permulaan.”

(Nadin Amizah – Beranjak Dewasa)

“The world opens itself before whose with noble hearts.”

(Genshin Impact)

PERSEMBAHAN

Sebagai bentuk rasa syukur yang teramat dalam, karya kecil ini kupersembahkan kepada:

Kedua orang tua tercinta,

Bapak Suparlan

Dan

Ibu Rohmah Maleha

Cahaya dalam setiap langkahku. Terima kasih untuk peluh, kasih sayang, serta doa-doa yang selalu mengudara demi melihatku menggapai cita-cita. Kalian adalah sumber kekuatan dan motivasi terbesar bagi penulis untuk tidak pernah menyerah dalam setiap langkah.

Almamater kebanggaan,

Universitas Lampung.

Dan pada akhirnya, untuk seseorang yang selalu berusaha berdiri tegak dan tak henti memperjuangkan mimpinya,

Diriku.

SANWACANA

Puji dan syukur penulis panjatkan ke hadirat Allah SWT., yang telah melimpahkan rahmat, taufik, serta hidayah-Nya sehingga skripsi dengan judul “Pengembangan *Front-End* Sistem Asesmen Talenta Mahasiswa Universitas Lampung Berbasis Next.js dengan Metode *Rapid Application Development*” dapat diselesaikan dengan baik. Penulis menyadari bahwa perjalanan menyusun skripsi ini tidak lepas dari doa, dukungan, dan bimbingan yang sangat berarti dari berbagai pihak. Oleh karena itu, dengan kerendahan hati dan rasa syukur yang mendalam, penulis ingin menyampaikan ucapan terima kasih sebesar-besarnya kepada:

1. Allah Subhanahu Wa Ta'ala, Sang Maha Pencipta dan Maha Pengasih. *Alhamdulillah rabbil 'alamin*, atas segala nikmat sehat, karunia, petunjuk, serta kemudahan yang senantiasa menyertai setiap langkah penulis dari awal masa perkuliahan hingga karya sederhana ini dapat terselesaikan dengan baik.
2. Kedua orang tua tercinta, Bapak Suparlan dan Ibu Rohmah Maleha. Terima kasih tak terhingga atas doa yang tiada putus, pengorbanan, dukungan moral maupun material, serta kasih sayang tulus yang tak ternilai harganya.
3. Prof. Dr. Ir. Lusmeilia Afriani, D.E.A., IPM., ASEAN Eng., selaku Rektor Universitas Lampung.
4. Dr. Hi. Ahmad Herison, S.T., M.T., selaku Dekan Fakultas Teknik Universitas Lampung, serta Dr. Eng. Helmy Fitriawan, S.T., M.Sc., selaku Dekan Fakultas Teknik pada masa bakti sebelumnya.
5. Ibu Herlinawati, S.T., M.T., selaku Ketua Jurusan Teknik Elektro, Universitas Lampung.
6. Ibu Yessi Mulyani, S.T., M.T., selaku Ketua Program Studi Teknik Informatika, Universitas Lampung.
7. Dr. Eng. Ir. Mardiana, S.T., M.T., IPM., selaku Pembimbing Utama dan Ir. Trisya Septiana, S.T., M.T., IPM., selaku Pembimbing Pendamping. Terima

kasih atas keikhlasan waktu, kesabaran, serta ilmu yang senantiasa diberikan untuk mengarahkan penulis dari awal penyusunan hingga skripsi ini rampung.

8. Ir. Gigih Forda Nama, S.T., M.T.I., IPM., selaku Dosen Penguji, atas segala diskusi, evaluasi, serta masukan yang sangat bermanfaat untuk menyempurnakan penelitian dan sistem yang penulis kembangkan.
9. Bapak Puput Budi Wintoro, S.Kom., M.T.I., selaku Dosen Pembimbing Akademik, atas segala nasihat, arahan, dan kepedulian yang membimbing penulis selama menjalani masa perkuliahan.
10. Bapak dan Ibu Dosen serta Staf di Program Studi Teknik Informatika dan Jurusan Teknik Elektro Universitas Lampung.
11. Pemerintah Republik Indonesia melalui Kementerian Pendidikan Tinggi, Sains, dan Teknologi atas program Kartu Indonesia Pintar (KIP) Kuliah, atas kesempatan emas serta dukungan finansial yang telah diberikan, sehingga penulis dapat menempuh pendidikan tinggi di Universitas Lampung.
12. Dr. Usep Syaipudin, S.E., M.S.Ak., dan Bapak Moh. Johan Pratama, S.Psi., M.Psi., selaku pengelola UPA-PKK Universitas Lampung.
13. Muhammad Azka Naufal, selaku rekan seperjuangan dan tim dalam pengembangan sistem ini, atas dedikasi, kerja keras, serta kolaborasi yang luar biasa. Sistem ini tidak akan terwujud tanpa sinergi dan jerih payah yang kita lalui bersama.
14. Sahabat-sahabat seperjalananku, Rifdah Fitriani Saharrudin, Nazma Hevano, dan Kenedy Ale, atas tawa, dukungan, dan kebersamaan yang meringankan langkah selama masa perkuliahan.
15. Keluarga besar Medinfo. Kepada Kak Fanny, atas segala dukungan yang diberikan. Kepada Shely Adtamara, partner dan sekretaris divisi yang luar biasa, sungguh sebuah kebanggaan bisa melangkah dan berproses bersama selama masa kepengurusan. Serta untuk Fatur, adik-adikku dan seluruh keluarga Laskar Medinfo Cemara, terima kasih telah menitipkan banyak cerita, warna, dan semangat dalam perjalanan perkuliahan penulis. Semoga kelak, kita dapat bertemu kembali dan meraih titik terbaik masing-masing.

16. Rekan-rekan pengurus dan keluarga besar Forum Komunikasi Bidikmisi/KIP Kuliah Universitas Lampung, atas ruang untuk bertumbuh, mengabdikan, serta segala pengalaman dan kenangan berharga yang telah terukir selama ini.
17. Kepada adikku tercinta, Alma Aulia. Terima kasih atas segala doa dan dukungan tulus yang selalu menyertai langkah kakak. Kehadiranmu adalah salah satu penguat kakak dalam meraih mimpi.
18. Seluruh teman-teman yang telah meluangkan waktunya dan bersedia membantu menjadi responden pengujian dalam penelitian ini.
19. Teruntuk diriku sendiri yang telah memilih untuk tetap berdiri tegak, bertahan, dan berjuang sepenuh hati untuk meraih mimpi hingga berhasil sampai di titik ini. Semoga setiap peluh dan lelah yang tercurah menjadi buah pencapaian yang manis di masa mendatang.
20. Serta kepada seluruh pihak yang tidak dapat penulis sebutkan satu per satu, namun telah memberikan bantuan, doa, dan kontribusi berarti dalam penyelesaian skripsi ini.

Semoga segala kebaikan dan keikhlasan yang telah diberikan kepada penulis sampai dengan terselesaikannya skripsi ini mendapat balasan yang sebaik-baiknya dari Tuhan Yang Maha Esa. Penulis menyadari bahwa di dalam skripsi ini masih terdapat banyak kekurangan. Oleh karena itu, penulis sangat terbuka terhadap segala saran dan kritik yang konstruktif demi penyempurnaan karya ini. Akhir kata, semoga skripsi ini dapat memberikan manfaat bagi pembaca dan pihak-pihak yang membutuhkan. Aamiin.

Bandar Lampung, 9 Juni 2026



Faris Ilham Hidayat
NPM.2215061118

DAFTAR ISI

	Halaman
ABSTRAK	ii
DAFTAR ISI	xiii
DAFTAR GAMBAR	xvi
DAFTAR TABEL	xx
DAFTAR SINGKATAN	xxi
DAFTAR LAMPIRAN	xxiii
I. PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	4
1.3 Tujuan Penelitian.....	4
1.4 Manfaat Penelitian	4
1.5 Batasan Masalah.....	5
1.6 Sistematika Penulisan	6
II. TINJAUAN PUSTAKA	7
2.1 Landasan Teori	7
2.1.1 UPA-PKK Universitas Lampung	7
2.1.2 Asesmen Talenta Mahasiswa	8
2.1.3 <i>9-Box Talent Matrix</i>	11
2.2 Landasan Teknologi	13
2.2.1 <i>Metode Rapid Application Development</i>	13
2.2.2 <i>Front-End Development</i>	15
2.2.3 Antarmuka Pengguna (<i>User Interface</i>)	16
2.2.4 Pengalaman Pengguna (<i>User Experience</i>)	16
2.2.5 <i>Eight Golden Rules of Interface Design</i>	17
2.2.5.1 Evaluasi Penerapan <i>Eight Golden Rules</i>	18
2.2.6 <i>Single Sign-On</i>	20
2.2.7 <i>Framework Next.js</i>	21

2.2.8	TypeScript	22
2.2.9	Tailwind CSS	23
2.2.10	<i>Unified Modeling Language</i>	24
2.2.10.1	<i>Use Case Diagram</i>	24
2.2.10.2	<i>Activity Diagram</i>	25
2.2.11	<i>System Usability Scale</i>	26
2.2.12	<i>Black-Box Testing</i>	28
2.2.13	Google Lighthouse	29
2.3	Penelitian Terkait.....	30
III.	METODE PENELITIAN	37
3.1	Waktu dan Tempat Penelitian.....	37
3.2	Jadwal Penelitian.....	37
3.3	Tim Penelitian	38
3.4	Alat dan Bahan Penelitian.....	39
3.4.1	Alat Penelitian.....	39
3.4.2	Bahan Penelitian.....	41
3.5	Tahapan Penelitian	41
3.5.1	<i>Requirements Planning</i>	42
3.5.1.1	Diskusi dengan Pengguna	42
3.5.1.2	Analisis Sistem Sebelumnya	43
3.5.1.3	Studi Dokumen	44
3.5.1.4	Identifikasi Kebutuhan Sistem	44
3.5.2	<i>User Design</i>	47
3.5.3	<i>Construction</i>	48
3.5.4	<i>Cutover</i>	49
IV.	HASIL DAN PEMBAHASAN	53
4.1	Hasil	53
4.1.1	Iterasi Pertama.....	53
4.1.1.1	<i>User Design</i>	53
4.1.1.2	<i>Construction</i>	78
4.1.2	Iterasi Kedua	120
4.1.2.1	<i>User Design</i>	121
4.1.2.2	<i>Construction</i>	129
4.1.3	Iterasi Ketiga	139
4.1.3.1	User Design.....	140
4.1.3.2	<i>Construction</i>	143

4.1.4	<i>Cutover</i>	146
4.1.4.1	<i>System Testing</i>	146
4.1.4.2	<i>User Acceptance Testing</i>	149
4.1.4.3	<i>Deployment Sistem</i>	156
4.1.4.4	<i>Performance Testing</i>	157
4.1.4.5	<i>Availability Testing</i>	158
4.2	Pembahasan.....	159
4.2.1	Analisa Hasil <i>System Testing</i>	159
4.2.2	Analisa Hasil <i>User Acceptance Testing</i>	160
4.2.2.1	Analisa Hasil <i>System Usability Scale</i>	160
4.2.2.2	Analisa Hasil Evaluasi <i>Eight Golden Rules</i>	161
4.2.3	Analisa Hasil <i>Performance Testing</i>	162
V.	SIMPULAN DAN SARAN	164
5.1	Simpan	164
5.2	Saran.....	165
	DAFTAR PUSTAKA.....	166

DAFTAR GAMBAR

	Halaman
Gambar 2.1 Alur Logika Asesmen Talenta Mahasiswa	10
Gambar 2.2 <i>9-Box Talent Matrix</i> Universitas Lampung	12
Gambar 2.3 Siklus Metode <i>Rapid Application Development</i>	14
Gambar 2.4 Interpretasi Skor <i>System Usability Scale</i>	28
Gambar 3.1 Tim Penelitian.	38
Gambar 3.2 Tahapan Penelitian.	41
Gambar 3.3 Proses Bisnis Sistem Sebelumnya.....	43
Gambar 4.1 Proses Bisnis Sistem Asesmen Talenta Mahasiswa.	54
Gambar 4.2 <i>Use Case Diagram</i> Sistem Asesmen Talenta Mahasiswa Unila.	55
Gambar 4.3 <i>Activity Diagram Login</i> Mahasiswa.	57
Gambar 4.4 <i>Activity Diagram</i> Pengerjaan Asesmen.	59
Gambar 4.5 <i>Activity Diagram Login</i> Admin.	60
Gambar 4.6 <i>Activity Diagram Dashboard</i> Statistik.	60
Gambar 4.7 <i>Activity Diagram</i> Daftar Mahasiswa.	61
Gambar 4.8 <i>Activity Diagram</i> Manajemen Pengguna.....	62
Gambar 4.9 <i>Activity Diagram</i> Manajemen Asesmen.	63
Gambar 4.10 <i>Activity Diagram</i> Log Aktivitas.	64
Gambar 4.11 <i>Activity Diagram</i> Manajemen Profil Pengguna.....	65
Gambar 4.12 <i>Mockup</i> Halaman Beranda.	66
Gambar 4.13 <i>Mockup</i> Halaman <i>Login</i>	67
Gambar 4.14 <i>Mockup</i> Halaman Detail Asesmen.	68
Gambar 4.15 <i>Mockup</i> Halaman Pengerjaan Asesmen Minat Karir.....	68
Gambar 4.16 <i>Mockup</i> Halaman Pengerjaan Asesmen Karakteristik Diri.	69
Gambar 4.17 <i>Mockup</i> Halaman Pengerjaan Asesmen Kesejahteraan Psikologis.	69

Gambar 4.18 <i>Mockup</i> Halaman Hasil.	70
Gambar 4.19 <i>Mockup</i> Halaman Statistik Publik.	71
Gambar 4.20 <i>Mockup</i> Halaman Tentang Kami.	71
Gambar 4.21 <i>Mockup</i> Halaman Kontak.	72
Gambar 4.22 <i>Mockup</i> Halaman Profil Mahasiswa.	73
Gambar 4.23 <i>Mockup</i> Halaman Statistik <i>Dashboard</i>	74
Gambar 4.24 <i>Mockup</i> Halaman Daftar Mahasiswa.	74
Gambar 4.25 <i>Mockup</i> Halaman Manajemen Asesmen.	75
Gambar 4.26 <i>Mockup</i> Halaman Manajemen Pengguna.	76
Gambar 4.27 <i>Mockup</i> Halaman Log Aktivitas.	77
Gambar 4.28 <i>Mockup</i> Halaman Profil Admin.	77
Gambar 4.29 Konfigurasi Variabel Lingkungan.	80
Gambar 4.30 Konfigurasi Tailwind CSS.	82
Gambar 4.31 <i>Source Code</i> Beranda.	83
Gambar 4.32 Tampilan Halaman Beranda.	84
Gambar 4.33 <i>Source Code Login</i>	85
Gambar 4.34 Tampilan Halaman <i>Login</i> Mahasiswa.	86
Gambar 4.35 <i>Source Code</i> Detail Asesmen.	87
Gambar 4.36 Tampilan Halaman Detail Asesmen.	88
Gambar 4.37 <i>Source Code</i> Pengerjaan Asesmen.	89
Gambar 4.38 Tampilan Halaman Pengerjaan Tes Minat Karier.	90
Gambar 4.39 Tampilan Halaman Pengerjaan Tes Karakteristik Diri.	90
Gambar 4.40 Halaman Pengerjaan Tes Kesejahteraan Psikologis.	91
Gambar 4.41 <i>Source Code</i> Penerjemahan Laporan Hasil.	92
Gambar 4.42 <i>Source Code</i> Fungsi <i>Render</i> PDF Hasil.	93
Gambar 4.43 Tampilan Halaman Hasil.	93
Gambar 4.44 <i>Source Code</i> Pengeksportan Dokumen PDF.	95
Gambar 4.45 <i>Source Code</i> Visualisasi Data Statistik.	95
Gambar 4.46 Tampilan Halaman Statistik Publik.	96
Gambar 4.47 <i>Source Code</i> Tentang Kami.	97
Gambar 4.48 Tampilan Halaman Tentang Kami.	98
Gambar 4.49 <i>Source Code</i> Kontak.	99

Gambar 4.50 Tampilan Halaman Kontak.....	100
Gambar 4.51 <i>Source Code</i> Validasi <i>Password</i>	101
Gambar 4.52 <i>Source Code</i> Navigasi Tab Profil Mahasiswa.	101
Gambar 4.53 Tampilan Halaman Profil Mahasiswa.	102
Gambar 4.54 <i>Source Code</i> Filter Data Berbasis <i>Role</i>	103
Gambar 4.55 <i>Source Code</i> Optimasi <i>Memoization</i> dan Fungsi Salin Grafik.....	104
Gambar 4.56 Implementasi <i>9-Box Matrix</i>	105
Gambar 4.57 Tampilan Halaman Statistik <i>Dashboard</i>	105
Gambar 4.58 Implementasi <i>9-Box Talent Matrix</i>	106
Gambar 4.59 <i>Source Code</i> Optimasi Pencarian dan Filter Adaptif.....	108
Gambar 4.60 <i>Source Code</i> Mutasi Pengelolaan Status Massal.....	109
Gambar 4.61 Tampilan Halaman Daftar Mahasiswa.	109
Gambar 4.62 <i>Source Code</i> Logika Evaluasi Status Tes.	111
Gambar 4.63 <i>Source Code</i> Arsitektur Mutasi Pembaruan Periode Asesmen.....	111
Gambar 4.64 Tampilan Halaman Manajemen Asesmen.	112
Gambar 4.65 Tampilan Edit Konten Asesmen.	112
Gambar 4.66 Tampilan Pengaturan Periode Asesmen.	113
Gambar 4.67 <i>Source Code</i> Manajemen Pengguna.....	114
Gambar 4.68 Tampilan Halaman Manajemen Pengguna.....	115
Gambar 4.69 <i>Source Code</i> Log Aktivitas.....	116
Gambar 4.70 Tampilan Halaman Log Aktivitas.....	117
Gambar 4.71 <i>Source Code</i> Logika Cakupan Unit Kerja Profil Admin.....	118
Gambar 4.72 <i>Source Code</i> Sinkronisasi Memori Lokal Profil Admin.	118
Gambar 4.73 Tampilan Halaman Profil Admin.	119
Gambar 4.74 <i>Activity Diagram</i> Pengerjaan Asesmen Iterasi Kedua.....	122
Gambar 4.75 <i>Activity Diagram</i> Manajemen Profil Mahasiswa Iterasi Kedua....	123
Gambar 4.76 <i>Activity Diagram</i> Daftar Mahasiswa Iterasi Kedua.....	124
Gambar 4.77 <i>Activity Diagram</i> Manajemen Asesmen Iterasi Kedua.	125
Gambar 4.78 <i>Mockup</i> Halaman Statistik Publik Iterasi Kedua.....	126
Gambar 4.79 <i>Mockup</i> Halaman Daftar Asesmen.	127
Gambar 4.80 <i>Mockup</i> Halaman Profil Mahasiswa Iterasi Kedua.	127
Gambar 4.81 <i>Mockup</i> Halaman Daftar Mahasiswa Iterasi Kedua.	128

Gambar 4.82 <i>Mockup</i> Halaman Manajemen Asesmen Iterasi Kedua.....	129
Gambar 4.83 <i>Source Code</i> Statistik Publik Iterasi Kedua.	130
Gambar 4.84 Tampilan Halaman Statistik Publik Iterasi Kedua.	131
Gambar 4.85 <i>Source Code</i> Daftar Asesmen.....	132
Gambar 4.86 Tampilan Halaman Daftar Asesmen.....	133
Gambar 4.87 <i>Source Code</i> Profil Mahasiswa Iterasi Kedua.....	134
Gambar 4.88 Tampilan Halaman Profil Mahasiswa Iterasi Kedua.....	135
Gambar 4.89 <i>Source Code</i> Daftar Mahasiswa Iterasi Kedua.....	136
Gambar 4.90 Tampilan Halaman Daftar Mahasiswa Iterasi Kedua.....	136
Gambar 4.91 Tampilan Modal Riwayat Daftar Mahasiswa.....	137
Gambar 4.92 <i>Source Code</i> Manajemen Asesmen Iterasi Kedua.....	138
Gambar 4.93 Tampilan Halaman Manajemen Asesmen Iterasi Kedua.....	139
Gambar 4.94 <i>Activity Diagram Login</i> Mahasiswa Iterasi Ketiga.....	140
Gambar 4.95 <i>Activity Diagram</i> Manajemen Profil Mahasiswa Iterasi Ketiga....	141
Gambar 4.96 <i>Mockup</i> Halaman <i>Login</i> Mahasiswa Iterasi Ketiga.....	142
Gambar 4.97 <i>Mockup</i> Halaman Profil Mahasiswa Iterasi Ketiga.....	143
Gambar 4.98 <i>Source Code Login</i> Mahasiswa Iterasi Ketiga.....	144
Gambar 4.99 Tampilan Halaman <i>Login</i> Mahasiswa Iterasi Ketiga.....	144
Gambar 4.100 <i>Source Code</i> Profil Mahasiswa Iterasi Ketiga.....	145
Gambar 4.101 Tampilan Halaman Profil Mahasiswa Iterasi Kedua.....	146
Gambar 4.102 Arsitektur <i>Deployment</i>	156
Gambar 4.103 Hasil Pengujian Performa Google Lighthouse.....	158
Gambar 4.104 Hasil Pemantauan <i>Dashboard</i> UptimeRobot.....	158

DAFTAR TABEL

	Halaman
Tabel 2.1 Komponen <i>Use Case Diagram</i>	24
Tabel 2.2 Komponen <i>Activity Diagram</i>	25
Tabel 2.3 Instrumen Pertanyaan <i>System Usability Scale</i>	27
Tabel 2.4 Penelitian Terkait.....	33
Tabel 3.1 Jadwal Penelitian.....	37
Tabel 3.2 Peran Tim Penelitian	38
Tabel 3.3 Alat yang Digunakan dalam Penelitian	39
Tabel 3.4 Bahan yang Digunakan dalam Penelitian.....	41
Tabel 3.5 Peran Pengguna Sistem	44
Tabel 3.6 Kebutuhan Fungsional Sistem.....	45
Tabel 3.7 Kebutuhan Non-Fungsional Sistem	46
Tabel 3.8 <i>Test Case</i> Pengujian <i>Black-Box</i>	49
Tabel 3.9 Butir Pertanyaan Evaluasi <i>Eight Golden Rules</i>	51
Tabel 4.1 Deskripsi <i>Use Case Diagram</i>	55
Tabel 4.2 Sampel Pengujian <i>Black-Box</i>	147
Tabel 4.3 Rekapitulasi Hasil Pengujian <i>Black-Box</i>	149
Tabel 4.4 Hasil Perhitungan Skor SUS	150
Tabel 4.5 Klasifikasi Kategori Penilaian <i>Eight Golden Rules</i>	153
Tabel 4.6 Rekapitulasi Penilaian <i>Eight Golden Rules</i>	153

DAFTAR SINGKATAN

Keterangan:

API	: <i>Application Programming Interface</i>
CSR	: <i>Client-Side Rendering</i>
CSS	: <i>Cascading Style Sheets</i>
CSV	: <i>Comma-Separated Values</i>
DOM	: <i>Document Object Model</i>
FAQ	: <i>Frequently Asked Questions</i>
FCP	: <i>First Contentful Paint</i>
HTML	: <i>HyperText Markup Language</i>
IdP	: <i>Identity Provider</i>
KF	: <i>Kebutuhan Fungsional</i>
KNF	: <i>Kebutuhan Non Fungsional</i>
LCP	: <i>Largest Contentful Paint</i>
MBTI	: <i>Myers-Briggs Type Indikator</i>
NPM	: <i>Nomor Pokok Mahasiswa</i>
PDF	: <i>Portable Document Format</i>
PWB	: <i>Psychological Well-Being</i>
RAD	: <i>Rapid Application Development</i>
RBAC	: <i>Role-Based Access Control</i>
SEO	: <i>Search Engine Optimization</i>
SSO	: <i>Single Sign-On</i>
SSR	: <i>Server-Side Rendering</i>
SUS	: <i>System Usability Scale</i>
TC	: <i>Test Case</i>

UI	: <i>User Interface</i>
UML	: <i>Unified Modeling Language</i>
UPA-PKK	: Unit Penunjang Akademik Pengembangan Karier dan Kewirausahaan
UX	: <i>User Experience</i>
VPS	: <i>Virtual Private Server</i>

DAFTAR LAMPIRAN

Halaman

Lampiran 1. Dokumentasi dan Hasil Pengujian Sistem..... **Error! Bookmark not defined.**

I. PENDAHULUAN

1.1 Latar Belakang

Universitas Lampung sebagai salah satu perguruan tinggi negeri di Indonesia telah memiliki unit-unit pembinaan mahasiswa yang komprehensif, meliputi Unit Penunjang Akademik (UPA) Bimbingan dan Konseling (BK), UPA Pengembangan Karier dan Kewirausahaan (PKK), Tim Pengembangan Potensi Mahasiswa (TP2M), serta Dosen Pembimbing Akademik (PA). Belum tersedianya data yang berperan sebagai *anchor point* atau landasan utama bagi unit pembinaan dalam memberikan intervensi kepada mahasiswa menyebabkan program-program yang dijalankan oleh masing-masing unit masih bersifat sporadis dan sulit diukur efektivitasnya. Kondisi ini menimbulkan tantangan dalam menentukan siapa mahasiswa yang perlu mendapatkan pembinaan, jenis intervensi yang paling sesuai, serta pola pembinaan yang efektif. Akibatnya, upaya pembinaan yang dilakukan belum sepenuhnya optimal dalam mendukung pengembangan potensi mahasiswa secara terarah dan berkelanjutan [1].

Konsepsi manajemen talenta (*Talent Management*) pertama kali diperkenalkan oleh McKinsey & Company melalui salah satu studi yang dilakukan pada tahun 1997. Kemudian pada tahun 2001, konsep ini diterbitkan dalam bentuk buku berjudul "*The War for Talent*" yang ditulis oleh Michaels, Handfield-Jones, dan Axelrod [2]. Dalam lingkungan organisasi yang kompetitif saat ini, manajemen talenta telah menjadi inisiatif strategis bagi organisasi untuk menarik, mengembangkan, dan mempertahankan karyawan berbakat guna mencapai keunggulan kompetitif [3]. Secara lebih komprehensif, manajemen talenta didefinisikan sebagai aktivitas dan proses yang melibatkan identifikasi sistematis terhadap posisi-posisi kunci yang memberikan kontribusi berbeda terhadap

keunggulan kompetitif berkelanjutan organisasi. Proses ini mencakup pengembangan kolam talenta (*talent pool*) yang terdiri atas sumber daya berpotensi tinggi dan berkinerja tinggi untuk mengisi peran-peran tersebut, serta pengembangan arsitektur sumber daya manusia yang berbeda untuk memfasilitasi pengisian posisi ini dengan individu yang kompeten dan memastikan komitmen berkelanjutan mereka terhadap organisasi [3].

Penerapan pemetaan talenta di dunia pendidikan, khususnya perguruan tinggi, masih relatif baru. Dalam konteks pendidikan tinggi, manajemen talenta dipandang sebagai *key enabler* bagi universitas untuk mencapai misi dan visi mereka melalui optimalisasi sumber daya manusia [2]. Universitas Lampung (Unila) melalui UPA-PKK mengadaptasi konsep *talent mapping* ke dalam sistem pendidikan tinggi di Indonesia. Program *Talent Mapping* UPA-PKK Unila diterapkan kepada mahasiswa baru dengan tujuan untuk membantu mereka mengenali potensi diri sejak dini, menentukan arah pengembangan diri, serta memilih kegiatan dan jalur karier yang sesuai dengan kekuatan dan kepribadian mereka [1].

Perbandingan antara penerapan *talent mapping* di perusahaan dan universitas menunjukkan perbedaan mendasar dalam tujuan dan pendekatannya. Di perusahaan, fokusnya adalah menempatkan orang yang tepat di posisi yang tepat (*right man in the right place*) untuk meningkatkan kinerja organisasi dan efektivitas bisnis. Sedangkan di universitas, fokusnya adalah mengenali potensi dan mengarahkan mahasiswa pada pengembangan diri yang optimal, bukan semata-mata pada produktivitas, tetapi pada pembentukan karakter (*character building*) dan kesiapan karier masa depan. Adaptasi konsep *talent mapping* di lingkungan pendidikan seperti Unila merupakan inovasi strategis yang tidak hanya memperkuat sistem pembelajaran, tetapi juga mendukung upaya mencetak lulusan yang kompeten dan berdaya saing [1]. Dengan mengadaptasi konsep *Talent Matriks* yang membagi mahasiswa ke dalam 9 kotak, data tersebut menjadi sumber pelaksanaan intervensi yang berbeda-beda sesuai dengan kebutuhan masing-masing mahasiswa.

UPA-PKK Unila telah memiliki laman resmi yang menyediakan layanan asesmen minat dan bakat, namun saat ini sistem tersebut telah dinonaktifkan. Penonaktifan ini dilakukan karena *website* sebelumnya belum sepenuhnya

mendukung optimalisasi pemanfaatan data hasil asesmen. Kendala mendasar terletak pada ketiadaan fitur visualisasi data dari hasil yang telah diperoleh. Absennya fitur ini menyebabkan data hasil asesmen hanya tersimpan sebagai arsip statis, sehingga pihak UPA-PKK terkendala dalam memantau sebaran profil talenta mahasiswa secara *real-time* dan komprehensif. Urgensi pengembangan sistem baru juga didorong oleh kebutuhan untuk mengadaptasi kerangka kerja baru yang belum ada di sistem sebelumnya, yaitu *9-Box Talent Matrix*. Penerapan konsep *9-Box Talent Matrix* yang dikembangkan UPA-PKK Universitas Lampung dengan instrumen asesmen yang telah diperbarui menuntut adanya sistem yang mampu memproses variabel-variabel yang ditentukan untuk menghasilkan luaran berupa posisi kuadran talenta mahasiswa.

Berdasarkan permasalahan tersebut, diperlukan sebuah aplikasi berbasis *website* yang mampu menyediakan sistem asesmen untuk membantu mahasiswa mengenali potensi diri mereka secara lebih mendalam serta menyediakan fitur visualisasi data dari hasil asesmen tersebut secara komprehensif. Melalui implementasi kerangka kerja *9-Box Talent Matrix*, sistem ini akan memproses hasil asesmen menjadi pemetaan kuadran talenta yang sesuai. Data visual tersebut kemudian menjadi acuan strategis bagi universitas untuk merancang intervensi dan pola pembinaan yang spesifik, terukur, dan tepat sasaran bagi setiap mahasiswa.

Next.js diimplementasikan menjadi *framework* utama dalam pengembangan *front-end* situs web Asesmen Talenta Mahasiswa. Teknologi ini dipilih karena terbukti mampu menghasilkan aplikasi web yang responsif, efisien, dan mudah dikelola berkat dukungan fitur *file-based routing*, *Server-Side Rendering* (SSR), serta *Client-Side Rendering* (CSR) [4]. Selain pemilihan teknologi, penerapan metode *Rapid Application Development* (RAD) juga dinilai sangat relevan dalam pengembangan sistem ini. Metode RAD berfokus pada iterasi cepat, keterlibatan pengguna sejak tahap awal, serta fleksibilitas dalam merespons perubahan kebutuhan sistem [5]. Implementasi pendekatan ini memastikan seluruh kebutuhan mahasiswa dan pihak universitas terakomodasi secara presisi lewat pembuatan *prototype* yang dievaluasi serta disempurnakan secara berulang. Hal ini sejalan dengan kelebihan dari penerapan metode RAD yang mampu mempercepat siklus pengembangan perangkat lunak dan meningkatkan kepuasan pengguna akhir.

Dengan adanya sistem Asesmen Talenta Mahasiswa, universitas tidak hanya mampu meningkatkan efektivitas program pembinaan dan konseling, tetapi juga berperan aktif dalam membantu mahasiswa mengenali potensi terbaik mereka guna mencapai kesuksesan akademik dan karier di masa depan.

1.2 Rumusan Masalah

Berdasarkan latar belakang tersebut, rumusan masalah pada penelitian ini adalah bagaimana membangun *front-end* sistem Asesmen Talenta Mahasiswa menggunakan *framework* Next.js dengan metode RAD untuk memenuhi kebutuhan sistem?

1.3 Tujuan Penelitian

Tujuan yang ingin dicapai dari penelitian ini adalah menghasilkan *front-end* sistem Asesmen Talenta Mahasiswa menggunakan *framework* Next.js dengan menerapkan tahapan metode RAD untuk memenuhi kebutuhan sistem.

1.4 Manfaat Penelitian

Manfaat yang diharapkan dapat diberikan melalui penelitian ini adalah sebagai berikut:

1. Manfaat Teoritis

Penelitian ini diharapkan memberikan kontribusi dalam bidang rekayasa perangkat lunak, khususnya terkait penerapan *framework* Next.js dalam memvisualisasikan logika asesmen pada aplikasi berbasis web, serta efektivitas metode RAD dalam mengakomodasi kebutuhan sistem.

2. Manfaat Praktis

a. Bagi Universitas Lampung

Hasil penelitian ini diharapkan dapat memfasilitasi identifikasi potensi dan arah karier mahasiswa secara lebih akurat. Melalui penyediaan *dashboard* visualisasi data talenta, sistem ini dapat mendukung unit-unit pembinaan dalam merumuskan strategi pembinaan yang tepat sasaran, terukur, dan berbasis data guna pengambilan keputusan strategis.

b. Bagi Mahasiswa

Penelitian ini memberikan kemudahan bagi mahasiswa dalam memetakan potensi diri, minat, serta arah karier yang relevan dengan kompetensi pribadi. Selain itu, sistem ini menawarkan pengalaman pengguna (*user experience*) yang interaktif dan mudah diakses, sehingga proses asesmen mandiri dapat dilakukan secara nyaman untuk mendukung perencanaan pengembangan diri.

1.5 Batasan Masalah

Adapun batasan masalah dalam penelitian ini adalah sebagai berikut:

1. Penelitian ini berfokus pada pembangunan antarmuka pengguna dan pengalaman pengguna aplikasi berbasis web menggunakan *framework* Next.js.
2. Pengguna sistem yang menjadi fokus penelitian terdiri atas dua pihak, yaitu mahasiswa baru Universitas Lampung serta pihak universitas sebagai administrator.
3. Logika asesmen yang diimplementasikan pada sistem terbatas pada model *9-Box Matrix* yang dikembangkan oleh UPA-PKK Unila. Penelitian tidak membahas validitas alat ukur psikologi tersebut, melainkan fokus pada digitalisasi dan visualisasinya.
4. Keamanan sistem dibatasi pada implementasi mekanisme autentikasi dan otorisasi di sisi *Front-End* melalui konsumsi token yang dihasilkan oleh *Back-End* dan integrasi *Single Sign-On* (SSO) Universitas Lampung.

5. Pengujian sistem dilakukan pada 4 aspek: *System Testing* untuk menguji fungsionalitas aplikasi, *User Acceptance Testing* untuk mengukur tingkat kepuasan dan kemudahan penggunaan sistem dari sisi *end-user*, *Performance Testing* menggunakan *tools* Google Lighthouse, serta *Availibity Testing* untuk mengukur ketersediaan sistem.

1.6 Sistematika Penulisan

Adapun sistematika penulisan tersebut adalah sebagai berikut:

BAB I : PENDAHULUAN

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

BAB II : TINJAUAN PUSTAKA

Bab ini membahas teori-teori yang menjadi dasar dan landasan dalam penelitian serta hasil penelitian terdahulu yang relevan sebagai pembanding dan acuan dalam pengembangan sistem.

BAB III : METODOLOGI PENELITIAN

Bab ini menjelaskan pendekatan dan metode yang digunakan dalam penelitian, yaitu metode *Rapid Application Development* serta menjelaskan rancangan sistem yang dibuat.

BAB IV : HASIL DAN PEMBAHASAN

Bab ini memaparkan hasil pengembangan sistem Asesmen Talenta Mahasiswa, termasuk implementasi antarmuka pengguna, alur interaksi asesmen, dan hasil pengujian sistem.

BAB V : KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan dari hasil penelitian yang dilakukan serta saran untuk pengembangan sistem lebih lanjut.

DAFTAR PUSTAKA

LAMPIRAN

II. TINJAUAN PUSTAKA

2.1 Landasan Teori

2.1.1 UPA Pengembangan Karier dan Kewirausahaan Universitas Lampung

Unit Penunjang Akademik Pengembangan Karier dan Kewirausahaan Universitas Lampung merupakan unit kerja strategis yang memiliki mandat utama untuk menjembatani kesenjangan antara dunia pendidikan tinggi dan dunia kerja. Unit ini bertugas menyiapkan lulusan yang kompeten sesuai dengan kebutuhan industri serta menumbuhkan minat dan budaya kewirausahaan di kalangan mahasiswa dan alumni [6].

UPA-PKK mengemban berbagai fungsi vital, meliputi pelaksanaan inventarisasi dan identifikasi kebutuhan dunia usaha, dunia industri, serta dunia kerja (DUDIKA). Unit ini berperan dalam peningkatan kapabilitas mahasiswa melalui pengembangan karier, fasilitasi kerja sama kewirausahaan, serta pemberian layanan informasi karier yang komprehensif.

Secara operasional, UPA-PKK melaksanakan tiga aktivitas utama, yaitu [6]:

1. Melakukan perencanaan, pelaksanaan, dan pelaporan penelusuran lulusan (*tracer study*) serta pendataan profil lulusan.
2. Membekali lulusan dengan kompetensi yang relevan agar mampu berkompetisi secara global, baik sebagai tenaga profesional maupun wirausaha.
3. Mendukung program pemerintah dalam menyelaraskan kurikulum pendidikan tinggi dengan kebutuhan pasar kerja serta mengakselerasi pertumbuhan wirausaha muda di lingkungan kampus.

Struktur organisasi UPA-PKK Universitas Lampung terdiri atas beberapa divisi di bawah kepemimpinan Kepala UPA. Adapun struktur divisi tersebut meliputi [6]:

1. Divisi *Tracer Study* dan Jaminan Mutu,
2. Divisi Pembinaan dan Konsultasi Karier,
3. Divisi Kewirausahaan dan Inkubator Bisnis Mahasiswa,
4. Divisi *Assessment Center* dan Layanan Psikologi, serta
5. Divisi Kemitraan dan *Internship*.

Dalam pengembangan sistem Asesmen Talenta Mahasiswa ini, kolaborasi dilakukan dengan Divisi *Assessment Center* dan Layanan Psikologi. Divisi ini bertanggung jawab atas perancangan instrumen psikometri dan alur asesmen talenta. Adapun pemangku kepentingan utama (*key stakeholders*) yang terlibat langsung dalam perancangan sistem ini adalah Dr. Usep Syaipudin, S.E., M.S.Ak. selaku Kepala UPA-PKK dan Moh. Johan Pratama, S.Psi., M.Psi. selaku Kepala Divisi *Assessment Center* dan Layanan Psikologi yang bertindak sebagai *Project Leader* dalam pengembangan instrumen asesmen ini.

2.1.2 Asesmen Talenta Mahasiswa

Manajemen talenta (*Talent Management*) telah menjadi topik sentral dalam strategi organisasi modern sejak diperkenalkannya “*The War for Talent*” oleh McKinsey & Company pada akhir tahun 1990-an. Konsep ini muncul dari pemahaman bahwa sumber daya manusia yang unggul merupakan aset paling kritis bagi keberhasilan organisasi. Manajemen talenta didefinisikan sebagai inisiatif strategis organisasi untuk menarik, mengembangkan, dan mempertahankan karyawan berbakat guna mencapai keunggulan kompetitif [3].

Collings dan Mellahi mendefinisikan manajemen talenta strategis sebagai aktivitas dan proses identifikasi sistematis terhadap posisi-posisi kunci yang memberikan kontribusi berbeda (*differentially contribute*) pada keunggulan kompetitif berkelanjutan suatu organisasi. Proses ini mencakup pengembangan kolam talenta (*talent pool*) yang terdiri atas individu berpotensi tinggi untuk

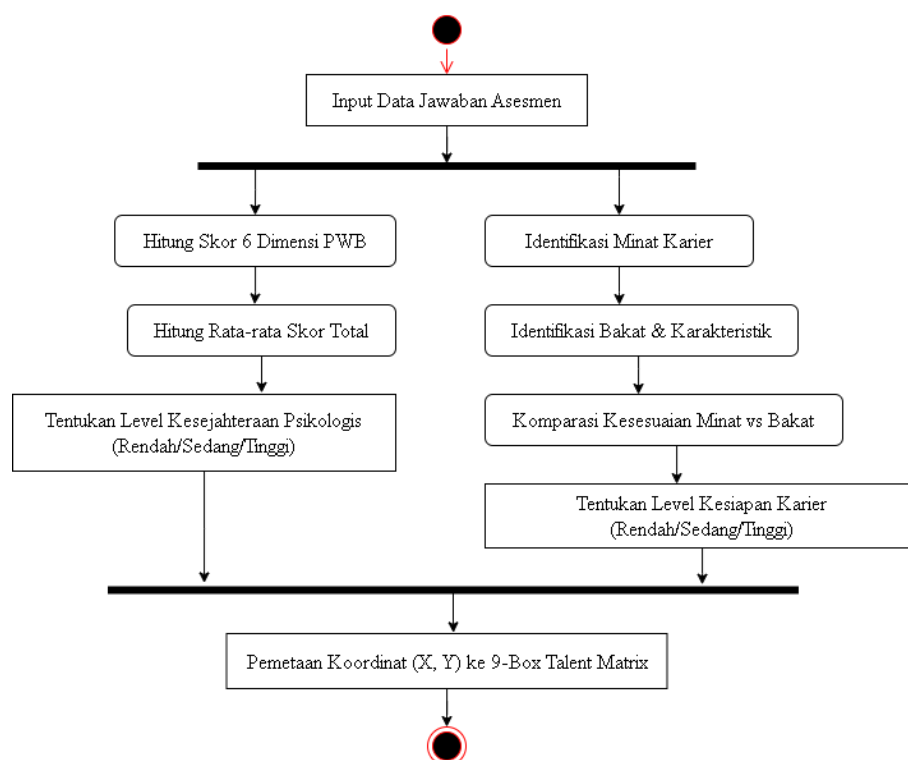
mengisi peran-peran strategis tersebut, serta penerapan arsitektur sumber daya manusia yang berbeda (*differentiated HR architecture*) untuk memastikan komitmen mereka terhadap organisasi [3]. Definisi ini menegaskan pergeseran fokus dari sekadar pengisian jabatan menjadi pengelolaan aset manusia yang bernilai strategis. Singh menambahkan bahwa urgensi manajemen talenta terletak pada dua alasan utama: memastikan organisasi mampu mengakuisisi dan mempertahankan talenta esensial, serta meningkatkan tingkat keterikatan (*engagement*) karyawan terhadap perusahaan [3].

Meskipun konsep manajemen talenta berakar pada sektor korporasi, penerapannya di sektor pendidikan tinggi kini semakin relevan. Razali dkk. mencatat bahwa manajemen talenta di perguruan tinggi dipandang sebagai mekanisme kunci (*key enabler*) untuk mencapai misi universitas melalui sumber daya manusianya [2].

Universitas Lampung melalui UPA-PKK mengembangkan Asesmen Talenta Mahasiswa dengan tujuan untuk membantu mereka mengenali potensi diri sejak dini, menentukan arah pengembangan diri, serta memilih kegiatan dan jalur karier yang sesuai dengan kekuatan dan kepribadian mereka. Pada sistem Asesmen Talenta Mahasiswa Universitas Lampung, kerangka asesmen diintegrasikan ke dalam dua dimensi utama [1].

Dimensi pertama dalam sistem ini berfokus pada identifikasi Bidang Karier Ideal, yang bertujuan mengklasifikasikan kecenderungan arah karier mahasiswa ke dalam empat domain utama, yakni: Praktisi, Akademisi, Pekerja Kreatif, dan Wirausaha. Struktur asesmen pada dimensi ini terbagi menjadi dua bagian. Bagian pertama difokuskan untuk mengidentifikasi preferensi minat karier, sedangkan bagian kedua dirancang untuk menelusuri bakat karier dan karakteristik personal menggunakan pendekatan *Myers-Briggs Type Indicator* (MBTI). Penerapan MBTI di sini berfungsi untuk memetakan pola perilaku dominan yang berkorelasi dengan keempat domain karier tersebut. Sebagai ilustrasi, tipe kepribadian yang cenderung analitis dan terstruktur diproyeksikan memiliki kesesuaian alami dengan domain Praktisi atau Akademisi. Analisis MBTI dalam asesmen ini juga dimanfaatkan untuk mengidentifikasi Gaya Komunikasi, Gaya Berpikir, dan Gaya Kerja guna memberikan gambaran karakteristik mahasiswa yang lebih komprehensif [1].

Dimensi kedua berfokus pada evaluasi kesehatan mental positif melalui pengukuran *Psychological Well-Being* (PWB). Konsep ini memandang kesejahteraan bukan sekadar ketiadaan gangguan mental, melainkan kehadiran kualitas positif dan kekuatan personal. Huppert menambahkan bahwa PWB mencakup perasaan positif dan kemampuan individu secara efektif dalam kehidupan sehari-hari [7]. Mengacu pada teori Ryff, kesejahteraan psikologis tidak hanya sekadar bebas dari penyakit, tetapi mencakup enam dimensi fungsi positif: penerimaan diri (*Self-Acceptance*), kemandirian (*Autonomy*), tujuan hidup (*Purpose in Life*), penguasaan lingkungan (*Environmental Mastery*), pertumbuhan pribadi (*Personal Growth*), dan hubungan positif dengan orang lain (*Positive Relations with Others*) [7]. Implementasi PWB dalam asesmen talenta ini digunakan untuk memastikan bahwa mahasiswa tidak hanya kompeten secara teknis, tetapi juga memiliki ketahanan psikologis (*resilience*) dalam menghadapi dinamika akademik.



Gambar 2.1 Alur Logika Asesmen Talenta Mahasiswa [1].

Luaran dari integrasi kedua dimensi tersebut divisualisasikan menggunakan adaptasi model *9-Box Talent Matrix*. Meskipun awalnya dikembangkan oleh McKinsey untuk dunia korporat guna memetakan performa vs potensi [8], model

ini dimodifikasi dalam konteks Universitas Lampung. Sumbu Y merepresentasikan Kesesuaian Minat Bakat (Potensi Karier) sedangkan Sumbu X merepresentasikan PWB (Kesejahteraan Psikologis). Matriks ini menghasilkan sembilan kotak (*boxes*) yang menjadi dasar segmentasi mahasiswa, di mana setiap kotak merekomendasikan strategi intervensi yang berbeda bagi unit pembinaan seperti UPA-BK, UPA-PKK, dan TP2M. Asesmen ini bertransformasi dari sekadar alat ukur menjadi peta jalan pengembangan diri (*self-development roadmap*) yang terukur dan berbasis data.

2.1.3 9-Box Talent Matrix

9-Box Matrix merupakan sebuah instrumen yang digunakan dalam manajemen talenta untuk mengidentifikasi serta mengevaluasi kinerja dan potensi pengembangan individu. Alat ini telah menjadi instrumen integral pada sistem manajemen talenta di berbagai organisasi. Metode pemetaannya mengklasifikasikan individu ke dalam sebuah matriks 3x3 yang menghasilkan sembilan kotak kategori [8].

Matriks ini memvisualisasikan posisi individu melalui dua sumbu evaluasi utama: sumbu horizontal (X) yang merepresentasikan kinerja saat ini (*performance*) dan sumbu vertikal (Y) yang memproyeksikan potensi masa depan (*potential*). Konsep ini bersifat adaptif dan telah dimodifikasi oleh berbagai organisasi, sesuai dengan kebutuhan spesifik masing-masing entitas [8].

Inovasi pengembangan model *9-Box Talent Matrix* oleh Universitas Lampung diterapkan khusus untuk memetakan potensi dan kinerja mahasiswa. Model ini bertujuan membantu pihak universitas dalam mengidentifikasi dan mengembangkan potensi mahasiswa secara terarah, baik dalam aspek akademik maupun non-akademik. Dalam konteks mahasiswa Unila, dua dimensi dimensi tersebut didefinisikan sebagai berikut [1]:

1. Sumbu Vertikal (Y): Tingkat Kesiapan Karier.

Sumbu ini mengukur sejauh mana minat dan bakat mahasiswa selaras pada empat bidang karier utama, yaitu: Praktisi, Akademisi, Wirausaha, dan

Pekerja Kreatif. Semakin tinggi skor pada sumbu ini, semakin jelas dan siap arah karier mahasiswa tersebut.

2. Sumbu Horizontal (X): Kesejahteraan Psikologis

Dimensi ini mengukur enam aspek PWB yang menjadi indikator kapasitas mahasiswa untuk bertahan dan berkembang dalam tekanan akademik maupun dunia kerja.

Berdasarkan perpotongan skor Kesiapan Karier (Tinggi/Sedang/Rendah) dan Kesejahteraan Psikologis (Tinggi/Sedang/Rendah), mahasiswa dikelompokkan ke dalam 9 kategori talenta (kotak). Klasifikasi ini berfungsi sebagai dasar bagi sistem untuk memberikan rekomendasi intervensi ke tiap-tiap kategori [1].



Gambar 2.2 9-Box Talent Matrix Universitas Lampung [1].

Adapun penjelasan dari tiap-tiap klasifikasi talenta mahasiswa tersebut ialah sebagai berikut:

1. *Critical Mismatch*: Mahasiswa dengan kinerja dan potensi rendah. Kelompok ini membutuhkan intervensi mendasar untuk meningkatkan kesadaran diri dan motivasi belajar guna mencegah kegagalan studi.
2. *Underdeveloped Potential*: Mahasiswa dengan potensi yang belum tergali sepenuhnya, memerlukan dorongan motivasi dasar.
3. *Inconsistent Zone*: Mahasiswa dengan performa yang tidak stabil (fluktuatif) dan memerlukan pendampingan konsistensi.
4. *Latent Talent Zone*: Mahasiswa berpotensi tinggi namun belum menampilkan performa maksimal, memerlukan fasilitasi aktualisasi diri.
5. *Growth Zone*: Mahasiswa dengan performa cukup baik dan potensi berkembang, memerlukan dukungan lanjutan untuk optimalisasi.
6. *Happy But Misaligned*: Mahasiswa yang bersemangat (performa psikologis baik) namun arah minat/bakatnya belum sejalan dengan pengembangan yang diharapkan.
7. *Aligned Developers*: Mahasiswa dengan keseimbangan tinggi antara potensi dan kinerja, serta berperan aktif dalam kegiatan akademik/organisasi.
8. *Positive Explorers*: Mahasiswa yang menunjukkan kinerja baik, berorientasi pembelajaran, dan aktif mengembangkan diri.
9. *High-Fit Champions*: Mahasiswa dengan potensi dan kinerja terbaik. Mereka siap menjadi *role model*, mentor, dan kandidat utama program pengembangan lanjutan.

2.2 Landasan Teknologi

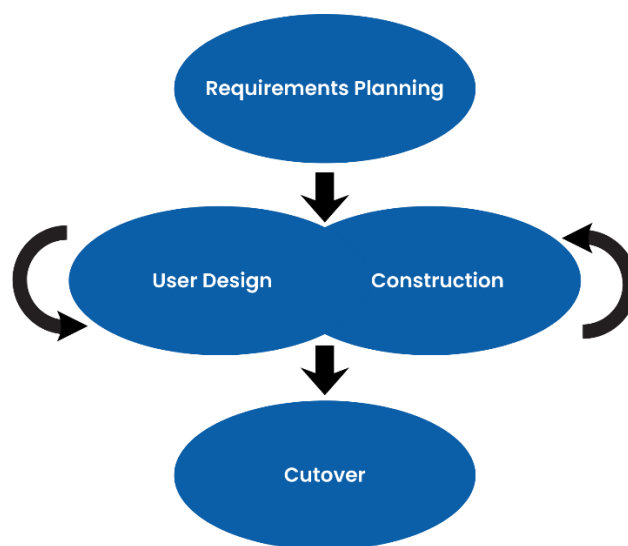
2.2.1 Metode *Rapid Application Development*

Rapid Application Development adalah sebuah teknik berbasis tim yang dirancang untuk mempercepat pengembangan sistem informasi guna menghasilkan sistem yang berfungsi secara cepat. Penggunaan metode ini bertujuan untuk menekan biaya pengembangan, mempersingkat waktu pengerjaan, serta meningkatkan probabilitas keberhasilan proyek. RAD sangat bergantung pada penggunaan *prototyping* dan keterlibatan pengguna secara intensif, di mana pengguna dapat

memeriksa model kerja sistem sedini mungkin untuk memastikan kesesuaian dengan kebutuhan mereka [5].

Metode RAD sangat relevan diterapkan pada pengembangan perangkat lunak kustom berskala besar yang membutuhkan waktu pengerjaan singkat. Pendekatan ini terbukti efektif dalam mengakomodasi implementasi algoritma spesifik pada suatu kasus kerja sekaligus memastikan skalabilitas sistem untuk pengembangan lanjutan [9].

Model RAD mengompresi siklus tradisional *System Development Life Cycle* (SDLC) menjadi empat fase pengembangan utama, meliputi [5]:



Gambar 2.3 Siklus Metode *Rapid Application Development* [5].

1. Perencanaan Kebutuhan (*Requirements Planning*)

Fase ini menggabungkan elemen perencanaan dan analisis sistem. Pengguna, manajer, dan staf IT berdiskusi untuk menyepakati kebutuhan bisnis, cakupan proyek, batasan, dan persyaratan sistem. Fase ini berakhir setelah tim mencapai kesepakatan dan memperoleh otorisasi manajemen.

2. Desain Pengguna (*User Design*)

Pengguna berinteraksi dengan analis sistem untuk mengembangkan model dan prototipe yang mewakili proses, *input*, dan *output* sistem. Proses ini bersifat interaktif dan berkelanjutan, memungkinkan pengguna untuk memodifikasi dan menyetujui model kerja yang sesuai dengan kebutuhan mereka.

3. Konstruksi (*Construction*)

Berfokus pada tugas pengembangan program dan aplikasi. Meskipun mirip dengan SDLC tradisional, pada fase RAD ini pengguna terus berpartisipasi untuk menyarankan perubahan atau perbaikan saat tampilan antarmuka atau laporan mulai dikembangkan.

4. Pelaksanaan (*Cutover*)

Merupakan fase akhir yang mencakup konversi data, pengujian, pergantian ke sistem baru, dan pelatihan pengguna. Dibandingkan metode tradisional, seluruh proses ini dilakukan secara ringkas sehingga sistem dapat segera dioperasikan.

Pemilihan metode RAD dalam pengembangan sistem ini didukung oleh kelebihan dari metode ini, yakni sistem dapat dikembangkan lebih cepat dengan penghematan waktu yang signifikan. Melalui keterlibatan pengguna sejak awal, tim dapat mendesain antarmuka sesuai kebutuhan secara lebih efektif. Modifikasi juga dapat dilakukan dengan cepat seiring perkembangan desain melalui keterlibatan pengguna.

2.2.2 Front-End Development

Front-end development merupakan bagian dari situs web yang berinteraksi langsung dengan pengguna. Bagian ini sering disebut sebagai sisi klien (*client-side*) dari sebuah aplikasi. Cakupan *front-end* mencakup semua elemen yang dialami pengguna secara langsung, termasuk warna dan gaya teks, gambar, grafik, tabel, tombol, serta menu navigasi. Pengembang *front-end* memanfaatkan tiga bahasa utama, meliputi HTML, CSS, dan JavaScript, untuk membangun struktur, desain, interaktivitas, dan konten visual pada layar peramban situs web, aplikasi web, maupun aplikasi seluler [10].

Terdapat dua tujuan utama dalam pengembangan sisi *front-end*, yaitu responsivitas (*responsiveness*) dan performa (*performance*). Seorang pengembang harus memastikan bahwa situs web bersifat responsif, yang berarti tampilan sistem harus muncul dengan benar pada berbagai ukuran perangkat. Hal ini krusial agar

tidak ada bagian dari situs web yang berperilaku abnormal terlepas dari dimensi layar yang digunakan oleh pengguna [10].

2.2.3 Antarmuka Pengguna (*User Interface*)

User Interface (UI) mendeskripsikan bagaimana pengguna berinteraksi dengan suatu sistem komputer. Antarmuka pengguna mencakup seluruh komponen perangkat keras, perangkat lunak, layar, menu, luaran (*output*), dan fitur pendukung yang menjembatani komunikasi dua arah antara pengguna dengan komputer. Kualitas UI memegang peranan krusial terhadap tingkat kebergunaan (*usability*) suatu sistem. Aspek ini secara langsung menentukan tingkat kepuasan pengalaman pengguna, kelancaran dukungan terhadap operasional bisnis organisasi, serta efektivitas pencapaian tujuan interaksi secara akurat [5].

2.2.4 Pengalaman Pengguna (*User Experience*)

User Experience (UX) merupakan proses desain yang bertujuan untuk meningkatkan kepuasan pengguna terhadap suatu produk dengan menciptakan interaksi yang berguna (*usable*), mudah diakses (*accessible*), dan menyenangkan antara produk dengan penggunanya. Desain UX memiliki cakupan yang lebih luas dibandingkan sekadar desain antarmuka pengguna, teknik ketergunaan (*usability*), atau rekayasa aksesibilitas. Fokus utama dari aktivitas ini adalah memberikan pengalaman positif bagi pengguna melalui serangkaian proses inkremental yang melibatkan pengembang serta para pemangku kepentingan [11].

Aspek pengalaman pengguna sangat krusial karena UX membentuk persepsi pengguna terhadap kualitas sebuah perangkat lunak. Keandalan komputasi dan kelengkapan fitur sebuah sistem tidak menjamin tingkat penerimaan yang tinggi apabila perangkat lunak tersebut rumit dioperasikan, rentan memicu kesalahan, atau menghalangi pencapaian tujuan pengguna. Perancangan UX wajib diintegrasikan sejak awal siklus hidup proyek, mengingat penyesuaian fungsionalitas antarmuka pada fase akhir pengembangan jarang berhasil menciptakan pengalaman interaksi yang optimal [11].

2.2.5 *Eight Golden Rules of Interface Design*

Eight Golden Rules merupakan salah satu panduan desain antarmuka dan metode evaluasi heuristik yang paling populer di dunia desain [12]. Panduan ini pertama kali diusulkan oleh Ben Shneiderman pada pertengahan tahun 1980-an dan telah mengalami revisi berkala untuk memastikan relevansinya dengan perkembangan teknologi [13].

Tujuan utama dari penerapan aturan-aturan ini adalah untuk membantu desainer dalam mengatasi berbagai masalah desain antarmuka serta menciptakan sistem yang ramah pengguna guna meningkatkan aspek *usability* sistem tersebut [14].

Adapun rincian kedelapan prinsip tersebut adalah sebagai berikut [13]:

1. *Strive for Consistency*

Antarmuka sistem harus mempertahankan konsistensi visual dan fungsional di seluruh halaman. Hal ini mencakup penggunaan jenis huruf, palet warna, terminologi menu, serta tata letak tombol yang seragam agar pengguna tidak merasa bingung saat berpindah halaman.

2. *Cater Universal Usability*

Desain antarmuka harus mampu mengakomodasi keragaman pengguna, baik pengguna pemula (*novice*) maupun pengguna ahli (*expert*). Sistem perlu menyediakan panduan bagi pemula, namun juga menyediakan fitur percepatan (*shortcuts*) bagi pengguna ahli untuk efisiensi kerja.

3. *Offer Informative Feedback*

Untuk setiap tindakan yang dilakukan pengguna, sistem wajib memberikan respon visual atau tekstual. Umpan balik ini bisa bersifat sederhana (seperti perubahan warna tombol saat diklik) atau bersifat krusial (seperti pesan konfirmasi “Data Berhasil Disimpan”) untuk memberikan kepastian kepada pengguna.

4. *Design Dialogs to Yield Closure*

Urutan aksi dalam sistem harus memiliki alur permulaan, pertengahan, dan akhir yang jelas. Umpan balik informatif pada akhir memberikan rasa lega

dan menandakan bahwa tugas telah tuntas, sehingga pengguna siap untuk aktivitas berikutnya.

5. *Prevent Errors*

Sistem sebaiknya dirancang sedemikian rupa agar pengguna tidak dapat membuat kesalahan fatal. Mekanisme ini dapat berupa validasi formulir otomatis yang mendeteksi *input* yang tidak sesuai sebelum data diproses, atau menonaktifkan tombol fungsi tertentu jika syarat belum terpenuhi.

6. *Permit Easy Reversal of Actions*

Sistem harus menyediakan fitur untuk membatalkan tindakan (*undo*) atau kembali ke langkah sebelumnya (*back*). Fitur ini mengurangi tingkat kecemasan pengguna saat mengeksplorasi sistem karena mereka mengetahui bahwa kesalahan dapat diperbaiki dengan mudah.

7. *Support Internal Locus of Control*

Pengguna harus ditempatkan sebagai inisiator aksi, bukan sekadar merespons perintah komputer. Sistem yang baik memberikan perasaan bahwa pengguna memegang kendali penuh atas antarmuka, menjadikan sistem sebagai alat yang responsif terhadap keinginan pengguna.

8. *Reduce Short-Term Memory Load*

Mengingat keterbatasan memori jangka pendek manusia, tampilan antarmuka harus dibuat sederhana. Informasi penting sebaiknya selalu ditampilkan atau mudah diakses kembali, sehingga pengguna tidak dibebani kewajiban untuk mengingat data dari satu tampilan ke tampilan lainnya.

2.2.5.1 Evaluasi Penerapan *Eight Golden Rules*

Verifikasi terhadap penerapan *Eight Golden Rules of Interface Design* pada suatu antarmuka sistem perlu dilakukan untuk mengukur tingkat penerimaan dan kepuasan pengguna. Evaluasi ini salah satunya dapat dilakukan melalui pendekatan kuantitatif menggunakan instrumen kuesioner. Metode pengukuran yang sering digunakan dalam evaluasi ini adalah skala Likert. Skala Likert merupakan skala yang digunakan untuk mengukur persepsi, sikap, atau pendapat pengguna terhadap

suatu topik, objek, atau fungsionalitas sistem tertentu melalui kriteria penilaian yang berjenjang [15].

Untuk mengukur seberapa baik penilaian responden terhadap implementasi desain antarmuka, diperlukan rangkaian perhitungan rentang nilai (*rating scale*) [15]. Penentuan skala interval penilaian pengalaman pengguna didasarkan pada perhitungan nilai terendah dan nilai tertinggi dengan formulasi sebagai berikut:

a. Penentuan Nilai Terendah

Nilai terendah merepresentasikan batas bawah dari total skor yang mungkin didapatkan jika seluruh responden memilih skor paling minimum pada instrumen [15].

$$N_{te} = r \times S_{min} \dots\dots\dots(2.1)$$

Keterangan:

N_{te} = Nilai Terendah

r = Jumlah Responden

S_{min} = Skor Minimum Skala Likert (misalnya: 1)

b. Penentuan Nilai Tertinggi

Nilai tertinggi merepresentasikan batas atas dari total skor yang mungkin didapatkan jika seluruh responden memilih skor paling maksimum pada instrumen [15].

$$N_{ti} = r \times S_{maks} \dots\dots\dots(2.2)$$

Keterangan:

N_{ti} = Nilai Tertinggi

r = Jumlah Responden

S_{maks} = Skor Maksimum Skala Likert (misalnya: 4)

c. Penentuan Rentang Kategori Penilaian

Panjang kelas interval digunakan untuk membagi jarak antara nilai tertinggi dan terendah menjadi lima tingkat penerimaan. Rumus hitung panjang interval adalah sebagai berikut [15]:

$$p = \frac{N_{ti} - N_{te}}{5} \dots\dots\dots (2.3)$$

Kategori penilaian pengalaman pengguna berdasarkan *Eight Golden Rules of Interface Design* kemudian diklasifikasikan ke dalam rentang nilai berikut:

- Sangat Tidak Baik: Nilai berada pada kuartil interval pertama.
- Tidak Baik: Nilai berada pada kuartil interval kedua.
- Kurang Baik: Nilai berada pada kuartil interval ketiga.
- Baik: Nilai berada pada kuartil interval keempat.
- Sangat Baik: Nilai berada pada kuartil interval kelima.

d. Teknik Penghitungan Skor Akhir

Tahap akhir dari proses evaluasi ini adalah menghitung total skor per butir pernyataan (T_s) dan rata-rata skor per prinsip desain. Total skor dihitung berdasarkan jumlah frekuensi jawaban responden dikalikan dengan bobot nilai masing-masing [15]:

$$T_s = \sum_{i=1}^n (F_i \times W_i) \dots\dots\dots (2.4)$$

Keterangan:

F_i = Frekuensi responden yang memilih skor tertentu.

W_i = Bobot nilai pilihan (1, 2, 3, atau 4).

2.2.6 Single Sign-On

Single Sign-On (SSO) adalah mekanisme autentikasi yang memungkinkan pengguna untuk mengakses berbagai aplikasi (*Relying Parties*) hanya dengan melakukan satu kali proses autentikasi. Mekanisme ini dirancang guna mengatasi kelelahan kata sandi (*password fatigue*) dengan menghilangkan kewajiban

pengguna dalam mengelola berbagai kredensial, yang secara langsung meningkatkan efisiensi pengalaman pengguna. Verifikasi identitas pada arsitektur ini dieksekusi secara terpusat oleh *Identity Provider* (IdP) [16].

2.2.7 Framework Next.js

Next.js merupakan *framework* React yang dirancang khusus untuk membangun aplikasi web *full-stack* secara efisien. Penggunaan teknologi ini mengolaborasikan komponen React untuk penyusunan antarmuka pengguna dengan kapabilitas Next.js dalam menyediakan berbagai fitur tambahan dan optimasi otomatis, termasuk konfigurasi alat tingkat rendah seperti *bundler* dan *compiler*. Hal ini memungkinkan pengembang, baik individu maupun tim, untuk lebih fokus pada pengembangan produk dan menghasilkan aplikasi React yang interaktif, dinamis, dan cepat [17].

Salah satu keunggulan utama Next.js adalah kemampuannya untuk memitigasi isu-isu pada aplikasi web tradisional dengan menggabungkan fitur terbaik dari SSR dan CSR [4]. Berbeda dengan aplikasi React standar yang umumnya hanya mendukung CSR, Next.js mendukung berbagai mekanisme pengambilan data yang dapat disesuaikan dengan kebutuhan aplikasi, antara lain [4]:

1. *Server-Side Rendering*: Mengonversi kode React menjadi HTML pada permintaan pertama untuk meningkatkan kecepatan akses.
2. *Static Site Generation*: Membangun halaman secara statis pada saat proses *build*.
3. *Incremental Static Regeneration*: Memungkinkan pembaruan konten statis secara bertahap tanpa perlu membangun ulang seluruh sistem.
4. *Pre-fetching*: Meningkatkan performa navigasi dengan memuat data halaman sebelum pengguna mengkliknya.

Next.js menawarkan beberapa keuntungan teknis yang signifikan bagi performa sistem informasi [18], di antaranya:

1. Optimasi SEO: Melalui strategi SSR, Next.js menyajikan HTML yang sudah siap dibaca oleh mesin pencari daripada mengirimkan beban Javascript yang berat ke peramban, sehingga berdampak positif pada peringkat pencarian.
2. Sistem *Routing* Sederhana: *Framework* ini mendukung mekanisme perutean berbasis *file-system* pada folder *pages* (atau *app*), sehingga tidak memerlukan pustaka pihak ketiga untuk mengelola navigasi antar halaman.
3. Kecepatan Pengembangan: Dengan optimasi otomatis dan komponen tautan (*Link*) bawaan, pengembang dapat melakukan pengalihan halaman secara efisien dan mempercepat waktu peluncuran produk.

2.2.8 TypeScript

TypeScript adalah bahasa pemrograman sumber terbuka (*open-source*) yang diperkenalkan oleh Microsoft pada tahun 2012 [19]. Bahasa ini beroperasi dalam format *superset* dari JavaScript, menjadikan setiap kode JavaScript yang valid secara otomatis diakui dan berfungsi penuh di dalam ekosistem TypeScript [20]. Perbedaan utamanya terletak pada implementasi pengecekan tipe statis (*static type checking*), di mana TypeScript akan melakukan pemindaian kode dan memunculkan peringatan jika ditemukan masalah tipe sebelum akhirnya memancarkan (*emit*) kode tersebut menjadi JavaScript standar [19][20].

TypeScript menyediakan berbagai fitur tingkat lanjut yang tidak dimiliki secara bawaan oleh JavaScript dinamis, seperti modul, *generics*, *classes*, dan *interfaces*. Salah satu dampak paling signifikan dari penggunaan TypeScript dalam pengembangan web adalah kemampuannya untuk mendeteksi kesalahan lebih dini. Sebagai bahasa dengan pengetikan statis, TypeScript dapat menangkap kesalahan sebelum *runtime* (saat aplikasi dijalankan), yang secara substansial mengurangi waktu dan tenaga yang dibutuhkan untuk proses *debugging* serta pengujian aplikasi. Hal ini memastikan kode yang dihasilkan lebih tangguh dan meminimalisir munculnya *bug* yang tidak terduga pada sisi pengguna [19].

2.2.9 Tailwind CSS

Tailwind CSS adalah sebuah *framework* CSS berbasis *utility-first* yang merevolusi pengembangan *front-end* dengan menawarkan cara yang sangat efisien dan dapat disesuaikan untuk menata gaya aplikasi web. Berbeda dengan *framework* tradisional seperti Bootstrap yang menyediakan komponen pradeфинisi, Tailwind CSS memungkinkan pengembang untuk menyusun gaya secara langsung di dalam HTML menggunakan kelas-kelas utilitas atomik yang kecil. Pendekatan ini secara signifikan meningkatkan kecepatan pengembangan, konsistensi, serta kemudahan dalam pemeliharaan kode karena pengembang dapat membangun desain kustom tanpa harus meninggalkan *file* HTML mereka [21].

Tailwind CSS memiliki beberapa fitur kunci yang mendukung efektivitas desain antarmuka, di antaranya [22][23]:

1. Pendekatan *Utility-First*: Menggunakan kelas utilitas kecil dan atomik untuk membangun desain yang kompleks, yang memungkinkan pengembang melakukan pembuatan prototipe dan iterasi desain secara cepat.
2. Sangat Dapat Disesuaikan (*Highly Customizable*): *Framework* ini mudah diperluas dan disesuaikan melalui file konfigurasi, yang memungkinkan pengembang mengatur setiap aspek sistem desain, mulai dari palet warna hingga skala spasi sesuai kebutuhan spesifik proyek.
3. Desain Responsif: Memiliki pengubah responsif bawaan yang memudahkan pembangunan antarmuka untuk berbagai ukuran perangkat melalui kelas intuitif untuk mengelola tata letak pada titik henti (*breakpoints*) yang berbeda.
4. Optimasi Performa: Gaya yang tidak digunakan akan dihapus (*purged*) pada tahap produksi, sehingga menghasilkan *file* CSS yang lebih kecil dan performa yang setara dengan CSS buatan tangan.

Penggunaan Tailwind CSS memberikan keunggulan dalam hal kecepatan dan fleksibilitas. Dengan menggunakan kelas utilitas pradeфинisi langsung di dalam HTML, pengembang dapat menjaga konsistensi desain di seluruh aplikasi tanpa perlu menulis banyak kode CSS kustom yang sering kali sulit dipelihara dalam jangka panjang.

2.2.10 Unified Modeling Language

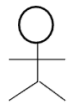
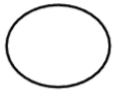
Unified Modeling Language (UML) merupakan metode yang digunakan secara luas untuk melakukan visualisasi dan dokumentasi desain sistem perangkat lunak. UML menerapkan konsep desain berorientasi objek dengan karakteristik independen terhadap spesifikasi bahasa pemrograman tertentu. Pemanfaatan bahasa pemodelan ini secara umum mencakup perancangan perangkat lunak, pendeskripsian alur proses bisnis, serta perumusan persyaratan sistem [5].

UML menyediakan berbagai alat grafis yang mendukung proses pemodelan kebutuhan, di antaranya diagram *Use Case* dan diagram *Activity*. Penggunaan UML dalam pengembangan sistem membantu memastikan bahwa arsitektur informasi dan aliran kerja yang kompleks dapat dipahami dengan baik oleh pengembang maupun pemangku kepentingan proyek [5].

2.2.10.1 Use Case Diagram

Diagram *use case* merupakan instrumen grafis untuk merepresentasikan interaksi visual antara pengguna dengan sistem informasi. Pemodelan ini memposisikan pengguna pada entitas *actor* yang memegang peran spesifik guna mendeskripsikan mekanisme interaksinya terhadap sistem [5]. Komponen utama dalam *Use Case Diagram* meliputi [24]:

Tabel 2.1 Komponen *Use Case Diagram*

No.	Simbol	Nama	Deskripsi
1		Aktor (<i>Actor</i>)	Merupakan entitas di luar sistem yang berinteraksi langsung dengan aplikasi, baik sebagai pengguna manusia maupun sistem eksternal lainnya.
2		<i>Use Case</i>	Merepresentasikan urutan aktivitas atau fungsi sistem yang dapat diamati dari luar untuk mencapai tujuan tertentu bagi aktor.

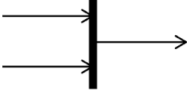
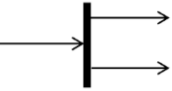
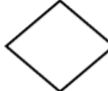
No.	Simbol	Nama	Deskripsi
3		<i>Association</i>	Menggambarkan jalur komunikasi dua arah antara aktor dan <i>use case</i> yang menandakan siapa yang melakukan apa.
4		<i>Include</i>	Menunjukkan bahwa sebuah <i>use case</i> wajib menyertakan fungsionalitas dari <i>use case</i> lain agar proses dapat berjalan.
5		<i>Extend</i>	Menunjukkan penambahan perilaku opsional pada <i>use case</i> dasar yang hanya dieksekusi pada kondisi tertentu.
6		<i>Generalization</i>	Hubungan spesialisasi di mana <i>use case</i> turunan mewarisi sifat dan perilaku dari <i>use case</i> induknya.

2.2.10.2 Activity Diagram

Activity Diagram berfungsi untuk menunjukkan tindakan dan peristiwa saat hal tersebut terjadi dalam suatu sistem. Diagram ini secara spesifik menampilkan urutan di mana tindakan tersebut berlangsung serta mengidentifikasi hasil atau luaran yang dihasilkan dari setiap tahapan proses [5]. Komponen yang terdapat dalam *Activity Diagram* meliputi [25]:

Tabel 2.2 Komponen *Activity Diagram*

No.	Simbol	Nama	Deskripsi
1		<i>Activity</i>	Merepresentasikan sebuah pekerjaan, tindakan, atau langkah operasional spesifik yang dilakukan dalam suatu sistem atau alur kerja.
2		<i>Control Flow</i>	Menunjukkan arah urutan eksekusi dari satu aktivitas ke aktivitas berikutnya.

No.	Simbol	Nama	Deskripsi
3		<i>Object Flow</i>	Menggambarkan aliran data atau objek yang dihasilkan dari satu aktivitas dan digunakan oleh aktivitas lainnya.
4		<i>Start point</i>	Menyatakan titik awal dimulainya suatu proses atau aliran aktivitas dalam diagram.
5		<i>End Point</i>	Menyatakan titik akhir di mana seluruh proses atau aliran aktivitas dalam sistem berhenti.
6		<i>Join</i>	Digunakan untuk menggabungkan kembali dua atau lebih aliran kegiatan paralel menjadi satu aliran tunggal.
7		<i>Fork</i>	Digunakan untuk memecah satu aliran aktivitas menjadi dua atau lebih aliran paralel yang dapat berjalan secara bersamaan.
8		<i>Decision</i>	Menunjukkan titik percabangan di mana sistem harus memilih satu jalur eksekusi berdasarkan kondisi tertentu.

2.2.11 System Usability Scale

System Usability Scale (SUS) merupakan sebuah skala yang bersifat “cepat dan praktis” (*quick and dirty*) untuk mengukur kegunaan (*usability*) suatu sistem. Instrumen ini dikembangkan sebagai alat yang efisien untuk menilai aspek ketergunaan sistem perangkat lunak secara keseluruhan dari sudut pandang pengguna [26]. Skala ini terdiri atas 10 butir pernyataan yang dijawab menggunakan skala Likert lima poin, mulai dari “Sangat Tidak Setuju” hingga “Sangat Setuju”. Pernyataan tersebut disusun secara bergantian antara pernyataan bernada positif dan negatif untuk menghindari bias jawaban dari responden. Berikut adalah 10 butir pernyataan dalam instrumen SUS [26]:

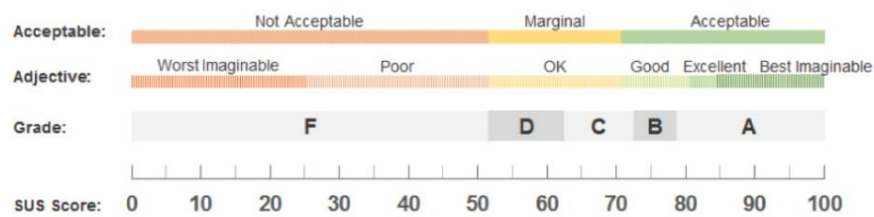
Tabel 2.3 Instrumen Pertanyaan *System Usability Scale*

No.	Pernyataan Instrumen SUS	Skala Likert (1-5)
1	Saya pikir saya akan sering menggunakan sistem ini.	1-5
2	Saya merasa sistem ini terlalu rumit padahal dapat dibuat lebih sederhana.	1-5
3	Saya rasa sistem ini mudah digunakan.	1-5
4	Saya pikir saya membutuhkan bantuan dari orang teknis untuk dapat menggunakan sistem ini.	1-5
5	Saya merasa berbagai fungsi dalam sistem ini terintegrasi dengan baik.	1-5
6	Saya merasa ada banyak ketidakkonsistenan dalam sistem ini.	1-5
7	Saya membayangkan kebanyakan orang akan dapat mempelajari sistem ini dengan sangat cepat.	1-5
8	Saya merasa sistem ini sangat membingungkan untuk digunakan.	1-5
9	Saya merasa sangat percaya diri saat menggunakan sistem ini.	1-5
10	Saya perlu mempelajari banyak hal sebelum saya dapat mulai menggunakan sistem ini.	1-5

Untuk memperoleh skor akhir SUS, diperlukan proses konversi dari posisi skala Likert (1-5) menjadi nilai kontribusi skor yang berkisar antara 0 hingga 5. Aturan perhitungan kontribusi skor dibedakan berdasarkan nomor butir pernyataan sebagai berikut [26]:

- Pernyataan Bernomor Ganjil (1, 3, 5, 7, dan 9): Nilai kontribusi skor didapat dari posisi skala dikurangi 1.
- Pernyataan Bernomor Genap (2, 4, 6, 8, dan 10): Nilai kontribusi skor didapat dari 5 dikurangi posisi skala.

Total penjumlahan dari seluruh kontribusi skor item dikalikan dengan besaran 2,5 untuk menghasilkan nilai akhir. Hasil akhir dari perhitungan ini akan menghasilkan nilai dengan rentang antara 0 hingga 100 yang merepresentasikan tingkat *usability* sistem secara keseluruhan.



Gambar 2.4 Interpretasi Skor *System Usability Scale* [27].

Skor SUS yang berkisar antara 0 hingga 100 diklasifikasikan ke dalam tiga kategori utama tingkat penerimaan, yaitu “*Not Acceptable*” (Tidak Dapat Diterima), “*Marginal*” (Marginal), dan “*Acceptable*” (Dapat Diterima). Skala ini juga menyertakan penilaian adjektiva yang membentang dari “*Worst Imaginable*” hingga “*Best Imaginable*” untuk menghubungkan skor kuantitatif dengan umpan balik subjektif pengguna. Rentang skor yang dianggap dapat diterima (*acceptable*) dimulai pada skor kurang lebih 70. Pada titik skor tersebut, kegunaan sistem dianggap telah bertransisi dari kategori “*OK*” menjadi “*Good*” atau lebih tinggi [27].

2.2.12 *Black-Box Testing*

Black-Box Testing adalah sebuah teknik pengujian perangkat lunak di mana cara kerja internal sistem (seperti kode, arsitektur, dan desain) tidak diketahui oleh penguji. Teknik ini berfokus pada evaluasi perilaku perangkat lunak melalui perspektif eksternal atau pengguna akhir (*end-user*). Pengujian fungsionalitas aplikasi dieksekusi murni tanpa meninjau struktur kode maupun memahami alur logika internal sistem. Pengujian dilakukan dengan membandingkan antara *input* dan *output* yang dihasilkan terhadap hasil yang diharapkan, jika terdapat ketidaksesuaian, maka ditemukan sebuah *bug* atau kesalahan pada sistem [28].

Metode *Black-Box Testing* menawarkan beberapa keuntungan dalam proses pengembangan perangkat lunak, antara lain [29]:

1. Eksekusi pengujian tidak memerlukan keahlian pemrograman atau pemahaman mendalam tentang struktur kode, sehingga dapat dilakukan oleh individu dengan berbagai tingkat keterampilan teknis.

2. Pengujian dijalankan dari sudut pandang pengguna akhir, sehingga memastikan fitur-fitur perangkat lunak selaras dengan kebutuhan pengguna dan spesifikasi yang telah ditetapkan.
3. Proses pengujian cenderung lebih cepat karena penguji tidak perlu melakukan analisis mendalam terhadap struktur kode internal.
4. Karena pengujian didasarkan pada spesifikasi perangkat lunak, dokumentasi yang dihasilkan menjadi lebih detail dan memudahkan pelacakan serta pengulangan tes di masa depan.

Salah satu teknik utama dalam *Black-Box Testing* adalah *Equivalence Partitioning*. Teknik ini membagi domain *input* dari sebuah program ke dalam kelas-kelas data untuk menurunkan kasus uji (*test cases*). Tujuannya adalah menciptakan kasus uji yang ideal dan efisien, di mana satu kasus uji saja mampu mengungkap satu kelas kesalahan tertentu (misalnya, kesalahan pemrosesan pada semua data karakter) yang jika tanpa teknik ini mungkin memerlukan eksekusi banyak kasus uji sebelum kesalahan tersebut terdeteksi [11].

2.2.13 Google Lighthouse

Google Lighthouse adalah alat *open-source* otomatis yang dikembangkan untuk membantu pengembang dalam meningkatkan kualitas halaman web. Alat ini memiliki kemampuan untuk melakukan serangkaian audit komprehensif yang mencakup aspek performa, aksesibilitas, *Search Engine Optimization* (SEO), dan *Best Practices* pada aplikasi web. Lighthouse bersifat fleksibel dan dapat dijalankan pada berbagai jenis halaman web, baik yang bersifat publik maupun yang memerlukan autentikasi [30].

Performa kecepatan memuat halaman (*page load speed*) merupakan faktor vital dalam keberhasilan aplikasi web seluler, di mana waktu muat melebihi 3 detik berisiko meningkatkan tingkat pengabaian pengguna hingga 53% dan menurunkan konversi bisnis secara signifikan. Metrik *Largest Contentful Paint* (LCP) diimplementasikan oleh Google menjadi standar pengukuran akurat yang menitikberatkan pada persepsi pengguna terhadap visibilitas konten utama di layar

(*viewport*). Mengacu pada standar *Core Web Vitals*, performa sistem dikategorikan “Baik” jika memiliki waktu muat LCP di bawah 2,5 detik, yang menjadi target teknis pengembangan guna menjamin responsivitas sistem dan kenyamanan pengalaman pengguna [31].

2.3 Penelitian Terkait

Tinjauan terhadap penelitian terkait dilakukan untuk memetakan posisi penelitian ini di antara pengembangan sistem serupa yang pernah dilakukan, serta untuk memastikan kebaruan dari sistem yang dikembangkan. Berikut adalah analisis terhadap penelitian yang relevan dalam kurun waktu lima tahun terakhir.

Penelitian pertama dilakukan oleh Bagas, Brata, dan Ananta [32] dengan judul “Pengembangan Sistem *Talent Mapping* untuk Memprediksi Karier dan Pemantauan Pengembangan Diri Mahasiswa (Studi Kasus: Direktorat Pengembangan Karier dan Alumni Universitas Brawijaya)”. Penelitian ini mengembangkan sistem berbasis web menggunakan teknologi Next.js dan TypeScript untuk sisi *front-end*, serta PostgreSQL sebagai basis data. Metode pengembangan yang digunakan adalah SDLC model *Prototyping* dengan integrasi algoritma logika *Fuzzy* untuk menghasilkan rekomendasi karier. Hasil pengujian menunjukkan bahwa penggunaan Next.js dan TypeScript berhasil lebih dari 90% kebutuhan fungsional berhasil diimplementasikan dan mencapai skor *usability* sebesar 85,69%.

Penelitian kedua oleh Setiabudhi, Wahyuni, dan Wibowo [33] berjudul “*Frontend Design and Development of Thesis Management Application Using NextJS in Physics Study Program UPN Veteran Jatim*” yang berfokus pada pembangunan antarmuka aplikasi manajemen skripsi menggunakan *framework* Next.js dan bahasa TypeScript. Penelitian ini menerapkan metodologi Scrum melalui enam iterasi *sprint*. Hasil pengujian *Black-Box* terhadap 177 persyaratan fungsional mendapatkan status *passed*, dan pengujian *User Acceptance Testing* menunjukkan bahwa seluruh fitur telah diterima oleh empat peran pengguna utama.

Penelitian ketiga dilakukan oleh M. F. Hasan, A. H. Brata, dan B. Priyambadha [34] dengan judul “Pengembangan Sistem Manajemen Data

Perencanaan Karir Mahasiswa Pada *Career Apps Dashboard* (Studi Kasus: Direktorat Pengembangan Karier Dan Alumni Universitas Brawijaya)”. Penelitian ini bertujuan untuk mengembangkan sistem berbasis web bernama “*Career Apps Dashboard*” guna mendukung pemetaan dan bimbingan karier berbasis data bagi mahasiswa. Pengembangan sistem menggunakan metode *SDLC Prototyping* yang dilakukan melalui 5 iterasi untuk menggali kebutuhan pengguna secara bertahap. Teknologi yang digunakan dalam implementasi sistem ini meliputi Next.js, Tailwind CSS, dan PostgreSQL. Hasil pengujian menunjukkan tingkat keberhasilan 100% pada uji *white-box*, *black-box*, dan *user acceptance*, serta mencapai *completion rate* sebesar 95% pada pengujian usabilitas.

Penelitian keempat oleh Ferdiansyah dkk. [35] dengan judul “Penerapan *Rapid Application Development* pada Pengembangan Perangkat Lunak *Talent dan Expert Management*” yang menerapkan metode RAD untuk membangun sistem manajemen talenta guna mendelegasikan proyek kepada dosen (*expert*) yang tepat. Hasilnya menunjukkan bahwa metode RAD efektif menciptakan sistem fungsional dalam waktu singkat (60-90 hari) dan memfasilitasi keterlibatan pengguna secara intensif melalui 15 fungsi utama yang berjalan dengan baik.

Penelitian kelima oleh Wibowo dan Narulita [36] dengan judul “*Development of a Web-based Quiz Management System using the Rapid Application Development Method*”. Penelitian ini berfokus pada rancang bangun sistem manajemen kuis berbasis *website* untuk memfasilitasi interaksi antara pengguna dan administrator dalam proses evaluasi digital secara efisien. Pengembangan sistem menerapkan metode RAD yang menekankan pada iterasi cepat dan keterlibatan pengguna melalui tahap identifikasi kebutuhan, perancangan antarmuka, dan pengujian. Perancangan sistem didokumentasikan menggunakan UML yang meliputi *Use Case Diagram*, *Activity Diagram*, dan *Entity Relationship Diagram*. Hasil penelitian menunjukkan bahwa sistem ini berhasil meningkatkan efisiensi evaluasi melalui antarmuka yang intuitif, sistem penilaian otomatis, serta fitur keamanan yang memastikan validitas data pengguna.

Penelitian keenam oleh P. D. Kharisma, M. Aminollah, dan A. K. Darmawan [37] dengan judul “Perancangan Aplikasi PSIKOTES dengan *Myers Briggs Type Indicator* (MBTI) dan *Big Five Personality Traits* Berbasis Web”. Penelitian ini

bertujuan untuk mendigitalisasi asesmen psikologis guna meningkatkan efisiensi dan relevansi rekomendasi karier bagi pengguna. Aplikasi ini dikembangkan menggunakan metode RAD yang bersifat iteratif untuk memastikan pengembangan yang cepat dan adaptif. Sistem yang dibangun menggabungkan dua metode kepribadian, yaitu MBTI dan *Big Five Personality Traits*, dengan fitur perhitungan hasil otomatis dan rekomendasi karier. Evaluasi sistem dilakukan menggunakan *PIECES Framework* untuk mengukur kualitas dari aspek performa hingga layanan.

Penelitian ketujuh merupakan studi internasional oleh Ismail dan Mansor [38] dengan judul “MyTalent: Pembangunan Sistem Pemetaan Bakat Berasaskan Web untuk Industri Kecil dan Sederhana dalam Kalangan Pelajar Siswazah” membangun sistem menggunakan model SDLC *Waterfall*. Sistem ini bertujuan membantu Industri Kecil dan Sederhana dalam mengenal pasti dan memetakan bakat pelajar pascasarjana agar sesuai dengan kebutuhan industri.

Penelitian kedelapan dilakukan oleh Luid, Bouty, dan Padiku [39] dengan judul “Sistem Pakar Untuk Perencanaan Karir Dengan Algoritma *Forward Chaining* Berbasis Web” mengembangkan aplikasi Careerpath menggunakan metode *Agile*. Penelitian ini memanfaatkan algoritma *Forward Chaining* untuk memberikan rekomendasi karier berdasarkan aturan (*rule-based*) yang ditanamkan dari pakar. Aplikasi ini bertujuan membantu siswa dalam merencanakan karier secara lebih relevan.

Penelitian kesembilan dilakukan oleh Triansyah dan Aji [40] dengan judul “Rancang Bangun Pengelompokan Bakat Kemampuan Bekerja Berbasis Desktop pada Telkomsigma Jakarta”. Penelitian ini bertujuan untuk mengotomatisasi proses pencarian dan pengelompokan bakat karyawan yang sebelumnya masih dilakukan secara manual. Sistem ini dibangun sebagai aplikasi desktop menggunakan bahasa pemrograman C# dengan bantuan *Integrated Development Environment* Visual Studio. Metode pengembangan perangkat lunak yang diterapkan adalah *Extreme Programming*. Hasil penelitian ini adalah sebuah aplikasi yang mampu mengelompokkan kemampuan kerja karyawan ke dalam sembilan kriteria penilaian untuk mendukung efisiensi manajemen talenta di perusahaan.

Penelitian terakhir dilakukan oleh M. D. R. A. Syamsuri, Tibyani, dan D. Pramono [41] dengan judul “Rancang Bangun Sistem Rekomendasi Karir Untuk

Siswa SMAN 2 Jombang Menggunakan Metode *Content-Based Filtering* Berdasarkan Nilai Tes Bakat Berbasis *Website*". Penelitian ini bertujuan untuk memberikan rekomendasi jalur karier yang akurat dan personal bagi siswa melalui pemanfaatan hasil tes bakat dan minat. Metode yang digunakan *adalah Content-Based Filtering* untuk mencocokkan profil siswa dengan deskripsi jurusan atau profesi yang relevan. Sistem ini dikembangkan berbasis web dan dievaluasi menggunakan metode SUS untuk mengukur tingkat kepuasan serta kemudahan penggunaan sistem oleh siswa.

Tabel 2.4 Penelitian Terkait

No.	Peneliti & Tahun	Judul Penelitian	Metode & Teknologi	Hasil & Relevansi
1	Bagas, Brata, dan Ananta (2025) [32]	Pengembangan Sistem <i>Talent Mapping</i> untuk Memprediksi Karier dan Pemantauan Pengembangan Diri Mahasiswa (Studi Kasus: Direktorat Pengembangan Karier dan Alumni Universitas Brawijaya)	SDLC model <i>Prototyping</i> , Next.js, TypeScript, PostgreSQL, Logika <i>Fuzzy</i>	Hasil: Implementasi >90% fitur fungsional dengan skor <i>usability</i> 85,69% . Relevansi: Memiliki kesamaan teknologi (<i>stack</i>) Next.js dan fokus pada topik <i>Talent Mapping</i> mahasiswa.
2	Setiabudhi, Wahyuni, dan Wibowo (2025) [33]	<i>Frontend Design and Development of Thesis Management Application Using NextJS in Physics Study Program UPN Veteran Jatim</i>	<i>Scrum</i> (6 iterasi <i>sprint</i>), Next.js, TypeScript	Hasil: Lulus uji <i>Black-Box</i> (177 persyaratan) dan diterima oleh seluruh peran pengguna melalui UAT. Relevansi: Relevan pada aspek pengembangan <i>Front-end</i> menggunakan Next.js di lingkungan akademik.

No.	Peneliti & Tahun	Judul Penelitian	Metode & Teknologi	Hasil & Relevansi
3	M. F. Hasan, A. H. Brata, dan B. Priyambadha (2025) [34]	Pengembangan Sistem Manajemen Data Perencanaan Karir Mahasiswa Pada <i>Career Apps Dashboard</i> (Studi Kasus: Direktorat Pengembangan Karier Dan Alumni Universitas Brawijaya)	SDLC <i>Prototyping</i> (5 iterasi), Next.js, Tailwind CSS, PostgreSQL	Hasil: Keberhasilan 100% pada uji fungsional dan mencapai <i>completion rate</i> usabilitas 95% . Relevansi: Kesamaan <i>stack</i> teknologi (Next.js, Tailwind CSS) dan fokus pada manajemen data karier mahasiswa.
4	Ferdiansyah dkk. (2023) [35]	Penerapan <i>Rapid Application Development</i> pada Pengembangan Perangkat Lunak <i>Talent dan Expert Management</i>	Metode RAD	Hasil: RAD efektif menciptakan sistem fungsional dalam waktu singkat (60-90 hari) dengan keterlibatan pengguna intensif . Relevansi: Kesamaan penggunaan metode pengembangan (RAD) dan topik manajemen talenta.
5	Hikmal Adi Wibowo & Siska Narulita (2025) [36]	<i>Development of a Web-based Quiz Management System using the Rapid Application Development Method</i>	Metode RAD, <i>Unified Modeling Language</i>	Hasil: Meningkatkan efisiensi evaluasi dengan antarmuka intuitif dan penilaian otomatis. Relevansi: Kesamaan metode RAD dan struktur sistem yang memisahkan <i>dashboard</i> admin dan <i>user</i>.

No.	Peneliti & Tahun	Judul Penelitian	Metode & Teknologi	Hasil & Relevansi
6	P. D. Kharisma, M. Aminollah, dan A. K. Darmawan (2025) [37]	Perancangan Aplikasi PSIKOTES dengan <i>Myers Briggs Type Indicator</i> (MBTI) dan <i>Big Five Personality Traits</i> Berbasis Web	Metode RAD, MBTI, <i>Big Five Personality Traits</i>	Hasil: Fitur perhitungan hasil otomatis dan rekomendasi karier, evaluasi menggunakan <i>PIECES Framework</i>. Relevansi: Sama-sama menggunakan metode RAD dan instrumen MBTI untuk asesmen potensi mahasiswa.
7	Ismail dan Mansor (2022) [38]	MyTalent: Pembangunan Sistem Pemetaan Bakat Berasaskan Web untuk Industri Kecil dan Sederhana dalam Kalangan Pelajar Siswazah	SDLC <i>Waterfall</i>	Hasil: Membantu memetakan bakat pelajar pascasarjana agar sesuai dengan kebutuhan industri. Relevansi: Kesamaan topik pemetaan bakat (<i>Talent Mapping</i>) berbasis web.
8	Luid, Bouty, dan Padiku (2024) [39]	Sistem Pakar untuk Perencanaan Karir Dengan Algoritma <i>Forward Chaining</i> Berbasis Web	<i>Agile</i> , algoritma <i>Forward Chaining</i>	Hasil: Memberikan rekomendasi karier yang akurat berdasarkan aturan pakar. Relevansi: Topik perencanaan karier mahasiswa berbasis web.
9	Triansyah dan Aji (2021) [40]	Rancang Bangun Pengelompokan Bakat Kemampuan Bekerja Berbasis <i>Desktop</i> pada Telkomsigma Jakarta	<i>Extreme Programming</i> , C#, Visual Studio	Hasil: Aplikasi mampu mengelompokkan bakat karyawan ke dalam sembilan kriteria penilaian secara otomatis.

No.	Peneliti & Tahun	Judul Penelitian	Metode & Teknologi	Hasil & Relevansi
				Relevansi: Relevan dalam penggunaan logika pengelompokan kriteria bakat.
10	M. D. R. A. Syamsuri, Tibyani, dan D. Pramono (2025) [41]	Rancang Bangun Sistem Rekomendasi Karir Untuk Siswa SMAN 2 Jombang Menggunakan Metode <i>Content-Based Filtering</i> Berdasarkan Nilai Tes Bakat Berbasis <i>Website</i>	<i>Content-Based Filtering, Web-based</i>	Hasil: Memberikan rekomendasi karier personal, pengujian usabilitas menggunakan SUS. Relevansi: Topik rekomendasi karier dan penggunaan metode evaluasi SUS.

III. METODE PENELITIAN

3.1 Waktu dan Tempat Penelitian

Waktu dan tempat pelaksanaan penelitian ini dilakukan pada:

1. Waktu Penelitian : Agustus 2025 sampai dengan Mei 2026
2. Tempat Penelitian : UPA-PKK Universitas Lampung

3.2 Jadwal Penelitian

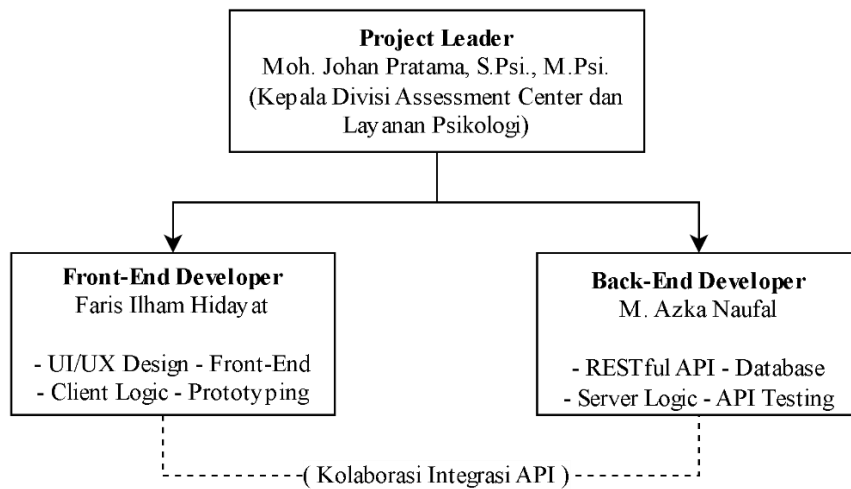
Jadwal pada penelitian ini adalah sebagai berikut:

Tabel 3.1 Jadwal Penelitian

[illegible]

3.3 Tim Penelitian

Proyek penelitian ini dikembangkan oleh sebuah tim yang dipimpin oleh pihak UPA-PKK Universitas Lampung. Tim ini mencakup seorang pengembang *front-end*, seorang pengembang *back-end*.



Gambar 3.1 Tim Penelitian.

Adapun rincian peran dalam tim penelitian ini adalah sebagai berikut:

Tabel 3.2 Peran Tim Penelitian

No.	Nama	Peran	Deskripsi Pekerjaan
1	Faris Ilham Hidayat	<i>Front-End Developer</i>	1. Merancang dan mengimplementasikan antarmuka pengguna menggunakan <i>framework</i> Next.js dan Tailwind CSS untuk menjamin responsivitas sistem di berbagai perangkat. 2. Mengembangkan fitur interaktif utama, khususnya visualisasi <i>Dashboard 9-Box Talent Matrix</i> dan halaman pengisian asesmen agar ramah pengguna. 3. Bertanggung jawab atas penerapan metode RAD pada sisi <i>client-side</i>, termasuk pembuatan prototipe dan revisi desain berdasarkan umpan balik pengguna.

No.	Nama	Peran	Deskripsi Pekerjaan
2	Muhammad Azka Naufal	<i>Back-end Developer</i>	1. Mengembangkan dan mengimplementasikan RESTful API menggunakan <i>framework</i> ElysiaJS sesuai dengan kebutuhan <i>front-end</i>. 2. Mengembangkan dan mengimplementasikan struktur tabel <i>database</i> untuk menyimpan data hasil asesmen, pengguna, dan soal asesmen. 3. Mengembangkan dan mengimplementasikan <i>business logic</i> penyimpanan serta perhitungan data tes asesmen ke dalam <i>9-Box Talent Matrix</i>, manajemen <i>database</i> pengguna, dan perhitungan statistik tes asesmen. 4. Melakukan kolaborasi dengan <i>front-end</i> untuk melakukan integrasi API pada sisi <i>client</i>. 5. Melakukan testing terhadap API yang telah dibuat.

3.4 Alat dan Bahan Penelitian

3.4.1 Alat Penelitian

Alat-alat yang digunakan dalam penelitian ini adalah sebagai berikut:

Tabel 3.3 Alat yang Digunakan dalam Penelitian

No.	Nama Alat	Spesifikasi	Deskripsi
1	Laptop	HP 14s-fq0021AU, Windows 11, AMD Ryzen 3, RAM 8GB	Perangkat keras utama yang digunakan untuk proses pengembangan sistem.
2	pnpm	Versi 10.22.0	<i>Package manager</i> yang digunakan untuk mengelola dependensi proyek dan menjalankan skrip.
3	Visual Studio Code	Versi Terbaru	<i>Integrated Development Environment</i> (IDE) utama untuk menulis, mengedit,

No.	Nama Alat	Spesifikasi	Deskripsi
			dan melakukan <i>debugging</i> kode program.
4	TypeScript	Versi 5.9.3	Bahasa pemrograman dengan pengetikan statis untuk meminimalisasi <i>bug</i> selama pengembangan.
5	Next.js	Versi 16.1.1	<i>Framework</i> utama berbasis React untuk membangun struktur aplikasi web, sistem <i>routing</i> , dan <i>rendering</i> .
6	Tailwind CSS	Versi 3.4.18	Kerangka kerja CSS <i>utility-first</i> untuk mempercepat proses penataan gaya antarmuka aplikasi.
7	Pustaka Eksternal (<i>Library</i>)	HeroUI, Headless UI, TanStack Query, React Hook Form, Zod, Recharts, React-PDF, jsPDF	Kumpulan pustaka tambahan untuk mempercepat pembangunan <i>website</i> .
8	Figma	Berbasis Web	Alat bantu untuk membuat desain <i>mockup</i> dan prototipe interaktif antarmuka pengguna.
9	Draw.io	Berbasis Web	Alat pemodelan untuk membuat diagram UML sebagai dokumentasi arsitektur sistem.
10	Git & GitHub	Versi Terbaru	Alat kontrol versi untuk menyimpan riwayat perubahan kode dan repositori.
11	Google Chrome	Versi Terbaru	Peramban utama untuk menjalankan aplikasi lokal serta melakukan pengujian fungsionalitas dan <i>debugging</i> .
12	Alat Uji Kinerja & Availabilitas	Google Lighthouse, UptimeRobot	Alat pengujian otomatis untuk memvalidasi <i>performance testing</i> (kecepatan sistem) dan <i>availability testing</i> .

3.4.2 Bahan Penelitian

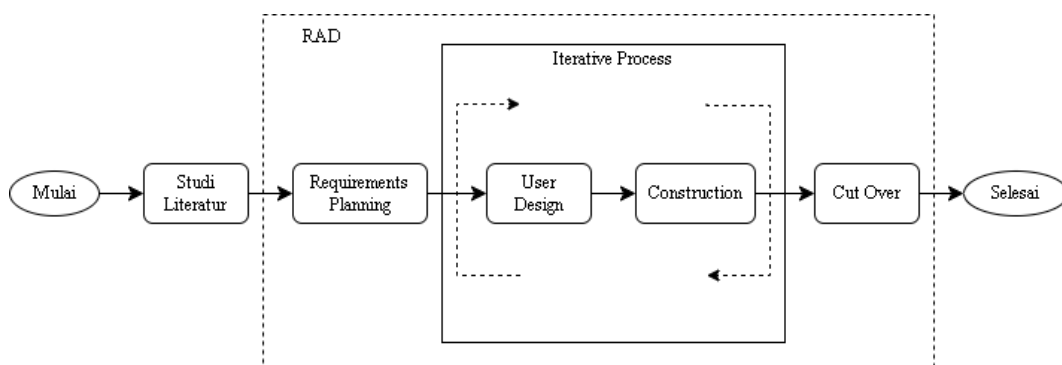
Bahan-bahan yang digunakan dalam pengembangan sistem ini adalah sebagai berikut.

Tabel 3.4 Bahan yang Digunakan dalam Penelitian

No.	Nama Bahan	Deskripsi	Fungsi
1	Instrumen Soal Asesmen	Butir pernyataan untuk modul Minat Karier, Karakteristik Diri dan Kesejahteraan Psikologis.	Sebagai konten utama yang akan ditampilkan pada antarmuka halaman pengisian asesmen.
2	Parameter Hasil dan 9-Box Talent Matrix	Aturan konversi jawaban asesmen serta <i>threshold</i> pemetaan 9 kategori talenta.	Sebagai acuan data yang diimplementasikan pada <i>client-side logic</i> untuk kalkulasi dan visualisasi hasil di <i>dashboard</i> .
3	Aset Identitas Visual	Logo Universitas Lampung, Dokumentasi, kode warna, dan tipografi resmi.	Sebagai <i>resource</i> mentah untuk menyusun <i>styling</i> antarmuka agar konsisten dengan standar institusi.

3.5 Tahapan Penelitian

Penelitian ini dilaksanakan menggunakan metode pengembangan RAD karena menekankan pada siklus pengembangan yang singkat dan keterlibatan pengguna secara aktif [5]. Tahapan penelitian dibagi menjadi empat fase utama, yaitu *Requirement Planning*, *User Design*, *Construction*, dan *Cutover* [5]. Alur tahapan penelitian dapat dilihat pada gambar berikut:



Gambar 3.2 Tahapan Penelitian.

3.5.1 *Requirements Planning*

Tahap ini merupakan fase awal yang krusial untuk menentukan arah pengembangan sistem. Pada tahap ini, peneliti berkolaborasi dengan pengguna, yaitu pihak UPA-PKK Universitas Lampung, untuk menyepakati lingkup dan tujuan sistem. Kegiatan spesifik yang dilakukan meliputi:

3.5.1.1 Diskusi dengan Pengguna

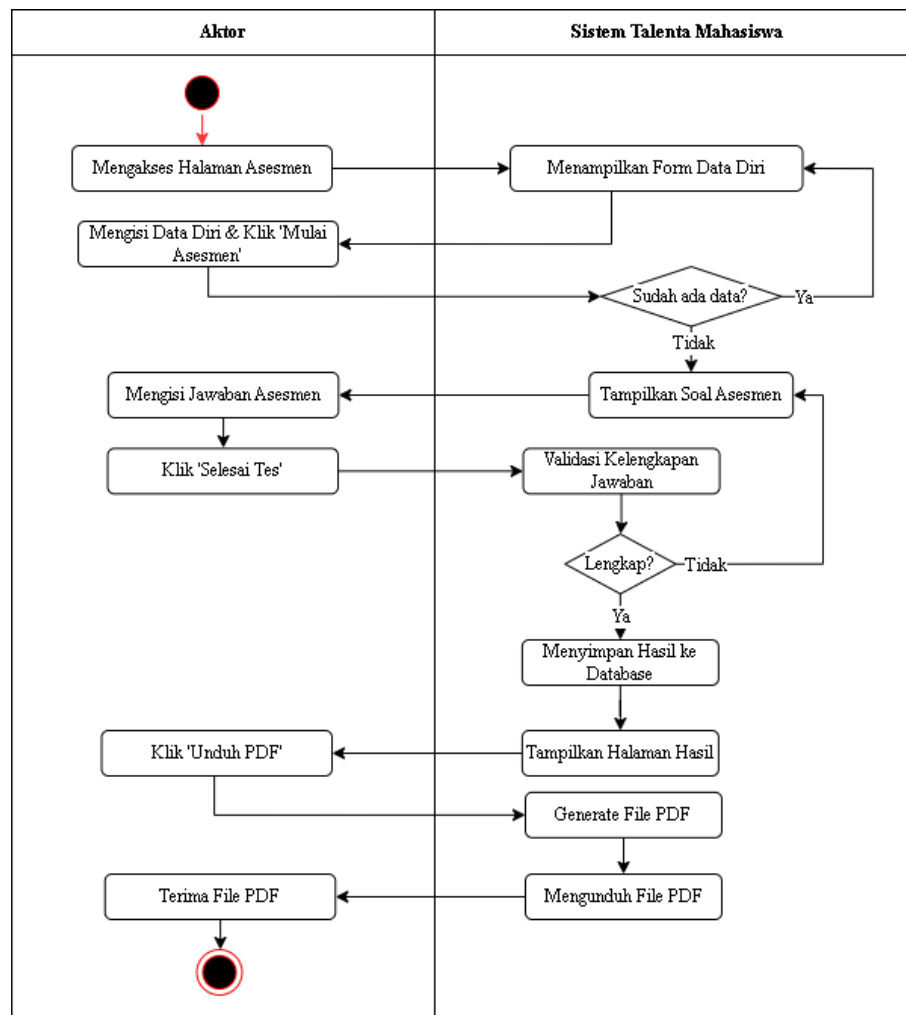
Langkah fundamental yang mengawali penelitian ini adalah pelaksanaan sesi tanya jawab dan diskusi melalui teknik wawancara semi-terstruktur dengan pihak UPA-PKK. Kegiatan ini bertujuan untuk memahami bagaimana alur asesmen yang diinginkan serta fitur-fitur apa saja yang dibutuhkan di dalam sistem. Hasil dari diskusi ini menjadi acuan utama dalam merancang sistem agar sesuai kebutuhan pihak pengguna.

Berdasarkan diskusi tersebut, ditetapkan beberapa parameter utama yang harus diakomodasi oleh sistem, antara lain:

1. Dibutuhkan sistem sebagai pusat pengembangan diri mahasiswa dengan fokus pada karier dan potensi diri.
2. Sistem akan mengelompokkan mahasiswa ke dalam bidang karier, yakni Praktisi, Akademisi, Pekerja Kreatif, atau Wirausaha.
3. Sistem dapat memvisualisasikan data tersebut ke dalam *dashboard* dan grafik.
4. Akan terdapat instrumen tes baru untuk sistem ini.
5. Mahasiswa dapat melakukan tes (satu kali pengisian) di dalam sistem, kemudian melihat hasil visualisasi data pribadi serta mengunduh laporannya dalam bentuk PDF.
6. Sistem menyediakan hak akses berjenjang untuk level Universitas, Fakultas, Jurusan, hingga Program Studi, di mana Administrator Universitas memiliki akses penuh terhadap seluruh data lintas fakultas, sedangkan tingkat di bawahnya (seperti Ketua Program Studi) dibatasi hanya pada rekapitulasi data mahasiswa di lingkup studinya masing-masing.

3.5.1.2 Analisis Sistem Sebelumnya

Analisis sistem sebelumnya dilakukan untuk memahami alur kerja dan identifikasi kendala pada sistem Asesmen Minat dan Bakat yang sebelumnya digunakan oleh UPA-PKK Universitas Lampung. Berikut adalah visualisasi proses bisnis pada sistem sebelumnya .



Gambar 3.3 Proses Bisnis Sistem Sebelumnya.

Alur operasional sistem pada Gambar 3.3 diawali dengan akses mahasiswa menuju halaman asesmen di laman resmi UPA-PKK Universitas Lampung. Absennya mekanisme autentikasi terintegrasi mengharuskan mahasiswa meng-*input* data diri secara manual sebagai prasyarat pengerjaan instrumen asesmen. Saat data diri mahasiswa sudah ada di dalam sistem, mahasiswa tidak dapat melakukan pengisian asesmen kembali. Kemudian, saat mahasiswa telah mengisi seluruh soal

yang tersedia, pengiriman seluruh jawaban akan tersimpan ke dalam basis data lalu menampilkan luaran langsung berupa laporan analisis profil talenta beserta rekomendasi pengembangan diri kepada mahasiswa yang dapat diunduh sebagai bahan referensi bagi mahasiswa ataupun pihak pengelola.

3.5.1.3 Studi Dokumen

Tahapan selanjutnya difokuskan pada studi dokumen dengan mempelajari secara rinci instrumen asesmen yang mencakup butir pernyataan Minat Karier, Karakteristik Diri, dan Kesejahteraan Psikologis, serta menganalisis parameter hasil asesmen untuk menentukan kriteria Minat Bakat Bidang Karier, Gaya Kerja, Gaya Komunikasi, Gaya Berpikir dan formulasi pemetaannya ke dalam model *9-Box Talent Matrix*.

3.5.1.4 Identifikasi Kebutuhan Sistem

Himpunan data awal menjadi landasan pelaksanaan tahap identifikasi kebutuhan guna mendefinisikan spesifikasi teknis sistem. Tahapan ini mengintegrasikan perumusan kebutuhan fungsional dan non-fungsional dengan analisis komprehensif terhadap peran pengguna demi menjamin keselarasan perangkat lunak terhadap kebutuhan operasional UPA-PKK.

a. Peran Pengguna

Struktur organisasi dan hierarki hak akses mendefinisikan 3 (tiga) peran *actor* utama yang berinteraksi langsung dengan sistem, meliputi:

Tabel 3.5 Peran Pengguna Sistem

No.	Pengguna	Deskripsi
1	Mahasiswa	Pengguna utama yang melakukan asesmen untuk mengetahui potensi diri.

No.	Pengguna	Deskripsi
2	Super Admin	Pengelola sistem dengan hak akses penuh untuk memantau data satu universitas dan mengelola data.
3	Admin (Universitas/Fakultas/Jurusan/ Program Studi)	Pemangku kepentingan yang memantau statistik dan laporan hasil mahasiswa sesuai dengan lingkup tingkatannya.

b. Kebutuhan Fungsional

Kebutuhan fungsional mendefinisikan layanan-layanan yang harus disediakan oleh sistem dan bagaimana sistem bereaksi terhadap *input* tertentu.

Tabel 3.6 Kebutuhan Fungsional Sistem

Kode	Nama Kebutuhan	Aktor	Deskripsi
KF-01	<i>Login & Autentikasi</i>	Mahasiswa, Admin, Super Admin	Sistem memverifikasi identitas pengguna menggunakan SSO Unila untuk mahasiswa dan kredensial khusus untuk aktor lainnya.
KF-02	Akses Halaman Publik	Publik, Semua Aktor	Sistem menampilkan informasi umum seperti <i>Homepage</i> , Statistik Umum, dan <i>About</i> tanpa mewajibkan <i>login</i> .
KF-03	Pengerjaan Asesmen	Mahasiswa	Sistem menyajikan modul soal asesmen dan menyimpan jawaban mahasiswa ke basis data.
KF-04	Lihat & Unduh Hasil	Mahasiswa, Admin, Super Admin	Sistem menampilkan hasil asesmen mahasiswa dan menyediakan fitur unduh dokumen hasil.
KF-05	Manajemen Asesmen	Super Admin	Sistem memungkinkan pengelolaan butir soal dan pengaturan akses waktu pengerjaan.
KF-06	Visualisasi Data Talenta	Admin, Super Admin	Sistem memvisualisasikan data talenta dalam bentuk grafik dan <i>9-Box Talent Matrix</i> sesuai wilayah akses serta menyediakan fitur filter.

Kode	Nama Kebutuhan	Aktor	Deskripsi
KF-07	Manajemen Daftar Mahasiswa	Admin, Super Admin	Sistem dapat menampilkan daftar data mahasiswa dan menyediakan fitur untuk pencarian, pemfilteran, peninjauan riwayat, ekspor data, serta fitur <i>reset</i> dan pemulihan jawaban mahasiswa.
KF-08	Mengelola Daftar Pengguna	Super Admin	Sistem menyediakan fitur untuk menambah, mengedit, menghapus, serta <i>me-reset password</i> pengguna di dalam sistem.
KF-09	Log Aktivitas	Super Admin	Sistem dapat mencatat dan menampilkan seluruh riwayat aktivitas yang dilakukan oleh pengguna di dalam sistem.
KF-10	Manajemen Profil Pengguna	Mahasiswa, Admin, Super Admin	Sistem memungkinkan aktor untuk mengubah dan mengelola informasi profil pribadi.
KF-11	<i>Logout</i>	Semua Aktor	Sistem mengakhiri sesi aktif pengguna dan mengembalikan tampilan ke halaman <i>login</i> demi keamanan data.

c. Kebutuhan Non-Fungsional

Kebutuhan non-fungsional mendefinisikan batasan layanan dan standar kualitas sistem, khususnya terkait dengan penggunaan teknologi Next.js.

Tabel 3.7 Kebutuhan Non-Fungsional Sistem

Kode	Parameter	Deskripsi Kebutuhan
KNF-01	<i>Performance</i>	Waktu muat halaman (<i>load time</i>) rata-rata di bawah 3 detik, didukung oleh fitur <i>Server-Side Rendering</i> Next.js untuk penanganan data visualisasi yang cepat.
KNF-02	<i>Security</i>	Integrasi dengan protokol keamanan SSO Universitas Lampung untuk autentikasi mahasiswa, serta penerapan <i>Role-Based Access Control</i> (RBAC) untuk membatasi akses data antar wilayah.

Kode	Parameter	Deskripsi Kebutuhan
KNF-03	<i>Responsiveness</i>	Antarmuka pengguna harus bersifat adaptif (<i>responsive</i>) dan dapat diakses dengan baik melalui perangkat <i>desktop</i> maupun <i>mobile smartphone</i> .
KNF-04	<i>Usability</i>	Alur pengerjaan asesmen dirancang linier untuk meminimalkan kebingungan pengguna (<i>user error</i>).
KNF-05	<i>Availability</i>	Sistem dapat diakses secara daring oleh pengguna melalui jaringan internet publik dengan <i>uptime</i> minimal 85%.

3.5.2 User Design

Tahap desain ini bertindak sebagai jembatan untuk menerjemahkan kebutuhan yang telah diidentifikasi ke dalam representasi visual dan struktural yang lebih konkret. Pada tahap *user design* dilakukan pemodelan proses bisnis, *use case diagram*, *activity diagram*, dan perancangan antarmuka. Pada tahap ini juga dilakukan sesi Evaluasi Prototipe secara partisipatif dengan pengguna. Rancangan antarmuka yang telah dibuat di Figma akan didemonstrasikan kepada pengguna, kemudian pengguna memberikan umpan balik (*feedback*) terkait alur navigasi dan tata letak. Masukan tersebut digunakan sebagai dasar perbaikan desain pada iterasi selanjutnya sebelum masuk ke tahap pengkodean.

1. Perancangan Proses Bisnis

Perancangan proses bisnis menjadi fondasi esensial guna menjamin keselarasan antara sistem yang dikembangkan dengan kebutuhan pengguna. Penelitian ini mengimplementasikan *activity diagram* sebagai instrumen utama untuk merancang sekaligus memodelkan alur operasional tersebut secara komprehensif.

2. Use Case Diagram

Use case diagram digunakan untuk menggambarkan fungsionalitas dari *website* Asesmen Talenta Mahasiswa dan interaksi antara pengguna (aktor) dengan sistem ini secara sistematis.

3. *Activity Diagram*

Activity diagram digunakan untuk memvisualisasikan urutan aktivitas dalam *website* Asesmen Talenta Mahasiswa serta mempermudah pemahaman proses sistem secara keseluruhan.

4. Perancangan *Mockup*

Pada perancangan ini dibuat sketsa halaman dari *website* Asesmen Talenta Mahasiswa dengan menerapkan prinsip *Eight Golden Rules* menggunakan alat bantu Figma sebelum diimplementasikan dalam bentuk kode.

3.5.3 *Construction*

Tahap konstruksi merupakan proses penerjemahan desain yang telah disetujui ke dalam baris kode program secara iteratif. Pada fase ini, pengguna terus berpartisipasi secara aktif untuk memberikan saran perubahan atau perbaikan saat tampilan antarmuka mulai dikembangkan guna memastikan kesesuaian sistem dengan kebutuhan pengguna. Adapun langkah-langkah dalam tahap ini meliputi:

1. Inisiasi Proyek

Melakukan instalasi dan konfigurasi lingkungan pengembangan menggunakan Next.js dengan bahasa TypeScript dan *styling* menggunakan Tailwind CSS.

2. Implementasi Antarmuka

Mengonversi desain menjadi komponen kode React yang dapat digunakan kembali (*reusable components*), seperti komponen tombol, *card*, dan modal.

3. Implementasi Logika Interaksi dan Visualisasi

Membangun logika validasi data di sisi klien (*client-side*) untuk memastikan kesesuaian data sebelum dikirim ke *server*. Pada tahap ini juga dikembangkan logika presentasi untuk menerjemahkan data hasil olahan server menjadi visualisasi yang informatif, meliputi pembuatan grafik profil PWB, pemetaan visual pada kuadran *9-Box Talent Matrix*, serta logika generasi dokumen laporan otomatis (PDF) yang dapat diunduh oleh pengguna.

4. Integrasi API

Berkolaborasi dengan *Backend Developer* untuk menghubungkan antarmuka *front end* dengan layanan API. Proses ini mencakup pengiriman data jawaban asesmen (*posting data*) dan pengambilan data hasil perhitungan (*fetching results*) untuk ditampilkan secara dinamis kepada pengguna.

5. Pengujian Unit

Melakukan pengujian mandiri terhadap komponen-komponen yang telah dibuat untuk memastikan fungsional sistem dan elemen antarmuka sesuai harapan.

3.5.4 Cutover

Tahap terakhir adalah implementasi sistem ke lingkungan produksi agar dapat digunakan oleh pengguna akhir. Kegiatan pada tahap ini meliputi:

1. Pengujian Sistem (*System Testing*)

Melakukan pengujian menyeluruh menggunakan metode *Black-Box Testing* dengan menggunakan teknik *equivalence partition* untuk memastikan seluruh fungsionalitas, mulai dari *login* hingga munculnya hasil berjalan tanpa kesalahan. *Test case* yang akan diuji pada sistem ini adalah sebagai berikut.

Tabel 3.8 *Test Case* Pengujian *Black-Box*

Kode	<i>Test Case</i>	Butir Uji
TC-01	Navigasi dan Antarmuka	Menguji perutean halaman, pembatasan hak akses berbasis peran, serta interaktivitas elemen antarmuka berjalan sesuai dengan spesifikasi desain sistem.
TC-02	Fitur Hubungi Kami	Menguji fungsi pengiriman pesan dari formulir antarmuka pengguna agar tersalurkan dengan tepat ke kontak pengelola UPA-PKK.
TC-03	Modul Autentikasi	Menguji proses masuk, pembatasan akses, manajemen kedaluwarsa, serta fungsionalitas proses keluar dari sistem.

Kode	<i>Test Case</i>	Butir Uji
TC-04	Alur Asesmen	Menguji kelancaran alur pengerjaan instrumen, mulai dari pemberian akses, proses pengisian, validasi kelengkapan data, hingga pemrosesan hasil akhir.
TC-05	Manajemen Profil Pengguna	Menguji fungsionalitas pembaruan data profil, perubahan kata sandi keamanan, serta peninjauan riwayat asesmen yang telah dilakukan oleh pengguna.
TC-06	<i>Dashboard Statistik</i>	Menguji tampilan visualisasi data agar menyesuaikan dengan lingkup hak akses akun, menguji fungsi filter data, interaktivitas grafik, serta fungsionalitas unduh laporan PDF.
TC-07	Daftar Mahasiswa	Menguji kesesuaian tampilan data mahasiswa berdasarkan lingkup hak akses, fungsi filter data, serta fitur aksi yang tersedia.
TC-08	Manajemen Asesmen	Menguji fungsionalitas pembuatan tes baru, penyuntingan konten tes, penghapusan data asesmen, serta konfigurasi pengaturan periode akses tes.
TC-09	Manajemen Pengguna	Menguji fungsionalitas penambahan, penyuntingan, pengaturan ulang <i>password</i> , dan penghapusan akun administrator.
TC-10	Log Aktivitas	Menguji sistem perekaman riwayat aktivitas, fungsi filter data rekaman log, serta fitur pengunduhan <i>file</i> laporan aktivitas.
TC-11	Logika Algoritma Hasil	Menguji keakuratan perhitungan dari jawaban asesmen untuk memastikan kesesuaian hasil pemetaan hasil.

2. Uji Penerimaan Pengguna (*User Acceptance Testing*)

Melakukan simulasi penggunaan aplikasi dengan pengguna, yakni mahasiswa dan pihak UPA-PKK, untuk mengukur tingkat kepuasan, kemudahan, dan penerimaan terhadap sistem. Pengujian penerimaan ini dilakukan menggunakan dua instrumen ukur, yaitu:

a. *System Usability Scale*

Digunakan untuk mengukur tingkat *usability* dan kemudahan penggunaan sistem secara keseluruhan.

b. Evaluasi *Eight Golden Rules*

Digunakan untuk mengukur tingkat penerimaan pengguna terhadap pengalaman dan interaksi antarmuka (*user interface*) sistem. Pengukuran dilakukan menggunakan kuesioner berskala Likert berdasarkan delapan prinsip desain antarmuka, yang kemudian dihitung nilai determinasi dan rentang rata-ratanya untuk menentukan skala pengalaman pengguna. Adapun rincian butir pernyataan yang digunakan sebagai instrumen evaluasi terhadap kedelapan prinsip antarmuka tersebut adalah sebagai berikut:

Tabel 3.9 Butir Pertanyaan Evaluasi *Eight Golden Rules*

No.	Prinsip	Pernyataan
1	<i>Strive for Consistency</i>	1. Penggunaan komposisi warna dan tata letak pada seluruh halaman <i>website</i> terlihat seragam dan menarik. 2. Susunan menu navigasi dan tombol aksi selalu berada pada posisi yang sama di setiap halaman sehingga tidak membingungkan.
2	<i>Cater Universal Usability</i>	1. Tampilan <i>website</i> sangat responsif dan tetap nyaman digunakan saat diakses melalui layar perangkat yang digunakan. 2. Fitur mode gelap/terang dan keterangan teks bantuan sangat membantu berbagai jenis pengguna dalam membaca informasi.
3	<i>Offer Informative Feedback</i>	1. <i>Website</i> selalu memberikan pesan pemberitahuan (sukses/gagal) yang jelas setiap kali pengguna selesai melakukan suatu tindakan. 2. Indikator <i>loading</i> (animasi berputar) sangat membantu memberi tahu pengguna bahwa sistem sedang memproses data.
4	<i>Design Dialogs to Yield Closure</i>	1. Alur penggunaan <i>website</i> dirancang dengan baik sehingga pengguna tahu persis langkah awal hingga akhir dari sebuah proses. 2. Pengguna selalu mendapatkan kepastian dan layar konfirmasi setelah menyelesaikan sebuah aktivitas penting di dalam sistem.
5	<i>Prevent Errors</i>	1. Sistem dengan baik mencegah terjadinya kesalahan dengan menonaktifkan tombol atau menahan proses jika data belum lengkap. 2. Peringatan kesalahan yang diberikan oleh sistem sangat jelas letaknya, sehingga pengguna tahu persis cara memperbaikinya.

No.	Prinsip	Pernyataan
6	<i>Permit Easy Reversal of Actions</i>	1. Pengguna dapat dengan mudah kembali ke halaman sebelumnya untuk membatalkan atau memperbaiki tindakan jika terjadi kesalahan. 2. Adanya kotak peringatan konfirmasi sebelum melakukan tindakan krusial sangat membantu mencegah kesalahan tekan yang tidak disengaja.
7	<i>Support Internal Locus of Control</i>	1. Pengguna merasa leluasa untuk berpindah dari satu halaman ke halaman lain tanpa merasa kebingungan, tersesat, atau dibatasi oleh sistem. 2. Pengguna merasa memegang kendali penuh dan merasa nyaman saat mengoperasikan berbagai fungsi di dalam <i>website</i> ini.
8	<i>Reduce Short-Term Memory Load</i>	1. Keterangan dan petunjuk pada antarmuka disajikan dengan jelas sehingga pengguna tidak perlu banyak mengingat instruksi. 2. Pilihan menu, tombol aksi, dan petunjuk selalu tersedia langsung di layar, sehingga pengguna cukup mengenalinya tanpa harus menghafal letak atau fungsinya.

3. *Deployment*

Mengunggah kode program ke layanan *Virtual Private Server* (VPS) dari Universitas Lampung agar sistem dapat diakses secara daring melalui internet.

4. Pengujian Performa (*Performance Testing*)

Pengujian ini dilakukan untuk memvalidasi kebutuhan non-fungsional terkait kecepatan waktu muat halaman (*load time*). Pengujian dilakukan menggunakan perangkat lunak bantu Google Lighthouse untuk mengukur metrik performa utama seperti FCP dan LCP guna memastikan sistem dapat dimuat dalam waktu kurang dari 3 detik sesuai spesifikasi yang ditetapkan.

5. Pengujian Ketersediaan (*Availability Testing*)

Pengujian ini dilakukan untuk memastikan keandalan sistem dalam melayani pengguna secara terus-menerus tanpa gangguan teknis yang berarti dengan memantau status server secara otomatis menggunakan layanan pihak ketiga, yaitu UptimeRobot. Pemantauan dilakukan melalui protokol HTTP/S dengan interval pemeriksaan setiap 5 menit guna mengukur persentase *uptime* dan mendeteksi adanya insiden *downtime*.

V. SIMPULAN DAN SARAN

5.1 Simpulan

Pengembangan *front-end* sistem Asesmen Talenta Mahasiswa Universitas Lampung menggunakan metode *Rapid Application Development* (RAD) dalam 3 (tiga) siklus iterasi telah berhasil diimplementasikan pada domain resmi pembinaantalent.unila.ac.id. Sistem ini berhasil memvisualisasikan instrumen pemetaan *9-Box Talent Matrix* yang didukung oleh 11 fungsionalitas, mencakup modul *login* dan autentikasi, akses halaman publik, pengerjaan asesmen, lihat dan unduh hasil, manajemen asesmen, visualisasi data talenta, manajemen daftar mahasiswa, manajemen pengguna, log aktivitas, manajemen profil pengguna, serta *logout*. Dari segi keandalan teknis, sistem mampu mencatatkan skor performa mencapai **99** pada pengujian Google Lighthouse dengan waktu muat konten utama hanya dalam 1,0 detik. Selain itu, pengujian fungsionalitas melalui 156 skenario *Black-Box* mencatatkan tingkat keberhasilan 100%, yang menandakan ketangguhan sistem dalam menangani berbagai kondisi operasional. Dari sisi pengalaman pengguna, sistem memperoleh skor *System Usability Scale* sebesar 75,87 (*Good*) dan evaluasi *Eight Golden Rules* dengan rata-rata 164,4 (Sangat Baik). Hasil ini menyatakan bahwa antarmuka sistem tidak hanya mudah digunakan dan dipelajari, tetapi juga memberikan pengalaman interaksi yang memuaskan bagi mahasiswa maupun administrator.

5.2 Saran

Adapun saran untuk pengembangan sistem Asesmen Talenta Mahasiswa selanjutnya adalah sebagai berikut:

1. Mengintegrasikan teknologi *Generative AI* atau *Large Language Model* (LLM) untuk mentransformasikan hasil asesmen menjadi narasi analisis kepribadian serta rekomendasi karier yang lebih mendalam, unik, dan personal bagi setiap individu mahasiswa.
2. Mengembangkan sistem web menjadi *Progressive Web App* (PWA) atau aplikasi mobile native untuk meningkatkan aksesibilitas bagi pengguna.
3. Menerapkan elemen gamifikasi pada antarmuka pengerjaan asesmen untuk meningkatkan pengalaman mahasiswa dalam mengisi asesmen.

DAFTAR PUSTAKA

- [1] UPA-PKK Universitas Lampung, “Model Pemetaan Talenta Mahasiswa Berbasis 9 Box Talent,” Bandar Lampung, 2025.
- [2] R. Razali, M. A. Arifin, L. A. Shafie, F. L. Azizan, dan M. A. M. Ishak, “Mapping the landscape of talent management research in higher education: a bibliometric analysis,” *Cogent Bus. Manag.*, vol. 11, no. 1, 2024, doi: 10.1080/23311975.2023.2298300.
- [3] R. P. Singh, “Talent Management Literature Review,” *Feed. J. Hum. Resour.*, vol. 1, no. 1, hal. 43, 2021, doi: 10.19166/ff.v1i1.3804.
- [4] S. Pati dan Y. Zaki, “Evaluating the Efficacy of Next.js: A Comparative Analysis with React.js on Performance, SEO, and Global Network Equity,” *WWW Companion 2025 - Companion Proc. ACM Web Conf. 2025*, vol. 2025, hal. 1239–1243, 2025, doi: 10.1145/3701716.3715565.
- [5] S. Tilley dan H. J. Rosenblatt, *Systems Analysis and Design*, 11 ed. Cengage Learning, 2016.
- [6] “Tentang CCED.” Diakses: 19 Oktober 2025. [Daring]. Tersedia pada: <https://cced.unila.ac.id/Site/About>
- [7] M. B. Mustafa *et al.*, “The Relationship between Psychological Well-Being and University Students Academic Achievement,” *Int. J. Acad. Res. Bus. Soc. Sci.*, vol. 10, no. 7, hal. 480–486, 2020, doi: 10.6007/ijarbss/v10-i7/7454.
- [8] A. Novia dan I. Yuadi, “Forecasting Employee Potential through Probationary Assessment,” *JBMP (Jurnal Bisnis, Manaj. dan Perbankan)*, vol. 9, no. 2, hal. 131–149, 2023, doi: 10.21070/jbmp.v9i2.1718.
- [9] Z. Pricillia, “Survey Paper : Perbandingan Metode Pengembangan Perangkat Lunak (Waterfall, Prototype, RAD),” *J. Bangkit Indones.*, vol. X, no. 01, hal. 6–12, 2021, doi: 10.52771/bangkitindonesia.v10i1.153.
- [10] P. D. Dutonde, “Website Development Technologies: A Review,” *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 10, no. 1, hal. 359–366, 2022, doi: 10.22214/ijraset.2022.39839.
- [11] R. S. Pressman dan B. R. Maxim, *Software Engineering: A Practitioner’s Approach*, 9th ed. New York: McGraw-Hill Education, 2019.
- [12] M. Benaïda, “Developing and extending usability heuristics evaluation for

user interface design via AHP,” *Soft Comput.*, vol. 27, no. 14, hal. 9693–9707, 2023, doi: 10.1007/s00500-022-07803-4.

- [13] J. Preece, Y. Rogers, dan H. Sharp, *Interaction Design: Beyond Human-Computer Interaction*, 5th ed. Indianapolis, IN: John Wiley & Sons, Inc., 2019.
- [14] Thamrin, G. Farell, P. Jaya, A. Hadi, dan S. Rahmadika, “User Interface Analysis on Job Matching Information System Using Eight Golden Rules,” hal. 160–168, 2023, doi: 10.2991/978-2-38476-056-5_18.
- [15] R. Nazli, N. Arbah, dan Jasri, “Analisis UI/UX Website Universitas Islam Kuantan Singingi Berdasarkan 8 Golden Rules of Interface Design,” *Technologica*, vol. 2, no. 2, hal. 61–69, 2023, doi: 10.55043/technologica.v2i2.101.
- [16] A. Zineddine, Y. Belfaik, A. Rehaime, Y. Sadqi, dan S. Safi, “Single Sign-On Security and Privacy: A Systematic Literature Review,” *Comput. Mater. Contin.*, vol. 84, no. 3, hal. 4019–4054, 2025, doi: 10.32604/cmc.2025.066139.
- [17] “Next.js Docs | Next.js.” Diakses: 25 Januari 2026. [Daring]. Tersedia pada: <https://nextjs.org/docs#what-is-nextjs>
- [18] H. H. Ben Kora dan M. S. Manita, “Modern Front-End Web Architecture Using React.js and Next.js,” *Univ. Zawia J. Eng. Sci. Technol.*, vol. 2, no. 1, hal. 1–13, 2024, doi: 10.26629/uzjest.2024.01.
- [19] S. Islam, “JavaScript alternative (TypeScript) and its effectiveness in web development,” Tampere University of Applied Sciences, 2023.
- [20] D. Vanderkam, *Effective TypeScript: 62 Specific Ways to Improve Your TypeScript*, 1st ed. Sebastopol, CA: O’Reilly Media, Inc., 2019.
- [21] W. A. Devidas, “Enhancing Web Accessibility with Tailwind CSS: A Framework Analysis,” *Int. J. Adv. Appl. Res.*, vol. 6, no. 23, hal. 15–21, 2025, doi: 10.5281/zenodo.15119118.
- [22] “Tailwind CSS Documentation.” Diakses: 25 Januari 2026. [Daring]. Tersedia pada: <https://tailwindcss.com/docs>
- [23] N. Kodali, “Tailwind CSS Integration in Angular : A Technical Overview,” *Int. J. Innov. Res. Sci. Eng. Technol.*, vol. 13, no. 9, 2024, doi: 10.15680/IJIRSET.2024.1309092.
- [24] Suharni, E. Susilowati, dan F. Pakusadewa, “Perancangan Website Rumah Makan Ninik Sebagai Media Promosi Menggunakan Unified Modelling Language,” *J. Rekayasa Inf.*, vol. 12, no. 1, hal. 1–12, 2023, [Daring]. Tersedia pada: <https://journal.istn.ac.id/index.php/rekayasainformasi/article/view/1527>
- [25] S. Ramdany, “Penerapan UML Class Diagram dalam Perancangan Sistem Informasi Perpustakaan Berbasis Web,” *J. Ind. Eng. Syst.*, vol. 5, no. 1, hal. 30–41, 2024, doi: 10.31599/2e9afp31.
- [26] J. Brooke, “SUS: A ‘Quick and Dirty’ Usability Scale,” *Usability Eval. Ind.*,

hal. 207–212, 2020, doi: 10.1201/9781498710411-35.

- [27] I. M. A. Sudestra, N. W. E. Agustini, I. M. A. O. Gunawan, G. Indrawan, dan M. Hakimi, “Improving Digital Learning: Evaluating the U Learn LMS With the System Usability Scale,” *JIPPI (Jurnal Ilm. Penelit. dan Pembelajaran Inform.*, vol. 9, no. 4, hal. 2325–2332, 2024, doi: 10.29100/jipi.v9i4.6910.
- [28] N. Golian, V. Golian, dan I. Afanasieva, “Black and White-Box Unit Testing for Web Applications,” *Bull. Natl. Tech. Univ. “KhPI”. Ser. Syst. Anal. Control Inf. Technol.*, vol. 0023, no. 1 (7), hal. 79–83, 2022, doi: 10.20998/2079-0023.2022.01.13.
- [29] A. Maspupah, “Literature Review: Advantages and Disadvantages of Black-Box and White Box Testing Methods,” *J. Techno Nusa Mandiri*, vol. 21, no. 2, hal. 151–156, 2024, doi: 10.33480/techno.v21i2.5776.
- [30] “Pengantar Lighthouse | Chrome for Developers.” Diakses: 1 Februari 2026. [Daring]. Tersedia pada: <https://developer.chrome.com/docs/lighthouse/overview?hl=id>
- [31] M. Kumar Dobbala, M. Shankar, dan S. Lingolu, “Web Performance Tooling and the Importance of Web Vitals,” *J. Technol. Innov.*, vol. 3, no. 3, hal. 1–15, 2022, [Daring]. Tersedia pada: <http://jtipublishing.com/jti>
- [32] B. Antarino, P. Sumaryunus, A. H. Brata, dan M. T. Ananta, “Pengembangan Sistem Talent Mapping untuk Memprediksi Karier dan Pemantauan Pengembangan Diri Mahasiswa (Studi Kasus: Direktorat Pengembangan Karier dan Alumni Universitas Brawijaya),” *J. Pengemb. Teknol. Inf. Dan Ilmu Komput.*, vol. 9, no. 10, hal. 1–9, 2025, [Daring]. Tersedia pada: <https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/15528>
- [33] A. S. Setiabudhi, E. D. Wahyuni, dan N. C. Wibowo, “Frontend Design and Development of Thesis Management Application Using NextJS in Physics Study Program UPN ‘Veteran’ Jatim,” *J. Teknol. Dan Open Source*, vol. 8, no. 1, hal. 142–152, 2025, doi: 10.36378/jtos.v8i1.4385.
- [34] M. F. Hasan, A. H. Brata, dan B. Priyambadha, “Pengembangan Sistem Manajemen Data Perencanaan Karir Mahasiswa Pada Career Apps Dashboard (Studi Kasus: Direktorat Pengembangan Karier Dan Alumni Universitas Brawijaya),” *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 9, no. 4, 2025, [Daring]. Tersedia pada: <http://j-ptiik.ub.ac.id>
- [35] F. R. Ferdiansyah, R. Sofian, D. F. Shiddieq, S. Budiarto, dan S. Utomo, “Penerapan Rapid Application Development Pada Pengembangan Perangkat Lunak Talent Dan Expert Management,” *Naratif J. Nas. Riset, Apl. dan Tek. Inform.*, vol. 5, no. 1, hal. 81–95, 2023, doi: 10.53580/naratif.v5i1.201.
- [36] H. A. Wibowo dan S. Narulita, “Development of a Web-based Quiz Management System using the Rapid Application Development Method,” vol. 18, no. 1, hal. 33–43, 2025, [Daring]. Tersedia pada: <http://journal.stekom.ac.id/index.php/pixel>
- [37] P. D. Kharisma, M. Aminollah, H. Aang, dan K. Darmawan, “Perancangan

Aplikasi PSIKOTES dengan Myers Briggs Type Indicator (MBTI) dan Big Five Personality Traits Berbasis Web,” *Semin. Nas. Hum. dan Apl. Teknol. Inf.*, vol. 11, no. 1, 2025.

- [38] M. A. H. Ismail dan M. Hj Mansor, “MYTALENT: Pembangunan Sistem Pemetaan Bakat Berasaskan Web Untuk Industri Kecil dan Sederhana dalam Kalangan Pelajar Siswazah,” *J. Eng. Technol. Appl. Sci.*, vol. 4, no. 3, hal. 140–153, 2022, doi: 10.36079/lamintang.jetas-0403.466.
- [39] M. F. J. K. Luid, A. A. Bouty, I. R. Padiku, dan Muthia, “Sistem Pakar Untuk Perencanaan Karir Dengan Algoritma Forward Chaining Berbasis Web,” *Diffus. J. Syst. Inf. Technol.*, vol. 4, no. 1, hal. 115–125, 2024.
- [40] J. Triansyah dan D. R. Aji, “Rancang Bangun Pengelompokan Bakat Kemampuan Bekerja Berbasis Desktop Pada Telkomsigma Jakarta,” *JIKA (Jurnal Inform.*, vol. 5, no. 1, hal. 16, 2021, doi: 10.31000/jika.v5i1.3191.
- [41] M. D. R. A. Syamsuri, Tibyani, dan D. Pramono, “Rancang Bangun Sistem Rekomendasi Karir Untuk Siswa SMAN 2 Jombang Menggunakan Metode Content-Based Filtering Berdasarkan Nilai Tes Bakat Berbasis Website,” *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 9, no. 6, 2025.