

**IMPLEMENTASI *SOFTWARE PRODUCT QUALITY* BERDASARKAN
MODEL ISO/IEC 25010 DALAM PENGUJIAN *WEBSITE* PT.INDONESIA
TEKNOLOGI PRENEUR (STUDI KASUS: DISNAKER KABUPATEN XYZ)**

(Skripsi)

Oleh

**NUR SETIOWATI
NPM 2017051013**



**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2024**

**IMPLEMENTASI *SOFTWARE PRODUCT QUALITY* BERDASARKAN
MODEL ISO/IEC 25010 DALAM PENGUJIAN *WEBSITE* PT.INDONESIA
TEKNOLOGI PRENEUR (STUDI KASUS: DISNAKER KABUPATEN XYZ)**

Oleh

NUR SETIOWATI

Skripsi

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

Pada

**Jurusan Ilmu Komputer
Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung**



**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2024**

ABSTRAK

IMPLEMENTASI *SOFTWARE PRODUCT QUALITY* BERDASARKAN MODEL ISO/IEC 25010 DALAM PENGUJIAN *WEBSITE* PT.INDONESIA TEKNOLOGI PRENEUR (STUDI KASUS: DISNAKER KABUPATEN XYZ)

Oleh

NUR SETIOWATI

Website Disnaker Kabupaten XYZ merupakan situs ketenagakerjaan milik salah satu kabupaten di provinsi Sumatera Selatan. Penelitian ini bertujuan mengimplementasikan model kualitas perangkat lunak ISO/IEC 25010 untuk menguji kualitas website Disnaker. Evaluasi dilakukan pada karakteristik *Functional Suitability*, *Performance Efficiency*, *Usability*, dan *Portability* dengan metode *Software Testing Life Cycle* (STLC). Pengujian dilakukan menggunakan berbagai alat seperti Selenium IDE, Katalon Recorder, JMeter, Maze, Useberry, dan BrowserStack. Hasil pengujian menunjukkan bahwa website Disnaker memiliki kualitas baik pada karakteristik *portability*, *performance efficiency*, dan *functional suitability*, namun karakteristik *usability* terindikasi membutuhkan peningkatan. Pada *Portability*, website menunjukkan kompatibilitas 99,01%, meski tidak mendukung Safari versi 13.1 pada MacOS Catalina. Karakteristik *fungsiional suitability* dengan sub-karakteristik *correctness* memperoleh persentase keberhasilan 87,23%, sub-karakteristik *appropriateness* 92,5% dengan indeks kepuasan "Sangat Memuaskan". Sedangkan sub-karakteristik *completeness* memperoleh nilai 0,95, mendekati 1, menandakan cakupan fungsional yang baik. *Usability* memperoleh skor 89% pada Maze dan 87% pada Useberry. Sementara itu, sistem menunjukkan performa waktu respons di bawah 10 detik tanpa *error*, mencerminkan efisiensi kinerja yang sangat baik. Hasil tersebut dapat digunakan sebagai perbaikan untuk peningkatan kualitas guna memenuhi kebutuhan pengguna serta standar perangkat lunak yang lebih tinggi.

Kata Kunci : Disnaker Kabupaten XYZ ; ISO/IEC 25010; Pengujian Perangkat Lunak; Kualitas Website.

ABSTRACT

IMPLEMENTATION OF SOFTWARE PRODUCT QUALITY BASED ON ISO/IEC 25010 MODEL IN WEBSITE TESTING PT.INDONESIA TEKNOLOGI PRENEUR (CASE STUDY: DISNAKER XYZ DISTRICT)

By

NUR SETIOWATI

The website Disnaker Kabupaten XYZ is a portal belonging to one of the districts in South Sumatra province. This study aims to apply the ISO/IEC 25010 software quality model to evaluate the quality of the Disnaker website. The evaluation focuses on the characteristics of Functional Suitability, Performance Efficiency, Usability, and Portability, using the Software Testing Life Cycle (STLC) method. Testing is conducted using various tools, such as Selenium IDE, Katalon Recorder, JMeter, Maze, Useberry, and BrowserStack. The test results show that the Disnaker website performs well in terms of portability, performance efficiency, and functional suitability, but the usability characteristic indicates that improvement is needed. Regarding portability, the website shows 99.01% compatibility, although it does not support Safari version 13.1 on macOS Catalina. The functional suitability characteristic, with the correctness sub-characteristic, achieved a success rate of 87.23%. The appropriateness sub-characteristic scored 92.5%, with a satisfaction index of 'Very Satisfactory.' Additionally, the completeness sub-characteristic scored 0.95, close to 1, indicating good functional coverage. Usability scored 89% on Maze and 87% on Useberry. Meanwhile, the system demonstrated a response time of under 10 seconds without errors, reflecting excellent performance efficiency. These results can be used to guide improvements for enhancing quality to meet user needs and higher software standards

Keywords: The Website Disnaker Kabupaten XYZ; ISO/IEC 25010; Software Testing; Website Quality.

Judul Skripsi : **IMPLEMENTASI SOFTWARE PRODUCT QUALITY BERDASARKAN MODEL ISO/IEC 25010 DALAM PENGUJIAN WEBSITE PT. INDONESIA TEKNOLOGI PRENEUR (STUDI KASUS: DISNAKER KABUPATEN XYZ)**

Nama Mahasiswa : **Nur Setiowati**

Nomor Pokok Mahasiswa : **2017051013**

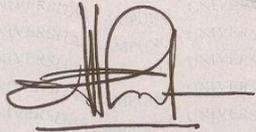
Program Studi : **S1 Ilmu Komputer**

Jurusan : **Ilmu Komputer**

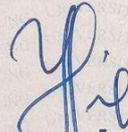
Fakultas : **Matematika dan Ilmu Pengetahuan Alam**

Menyetujui

1. **Komisi Pembimbing**

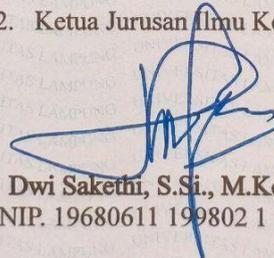


Dr. rer. nat. Akmal Junaidi, M.Sc.
NIP. 19710129 199702 1 001



Anie Rose Irawati, S.T., M.Cs.
NIP. 19791031 200604 2 002

2. **Ketua Jurusan Ilmu Komputer**

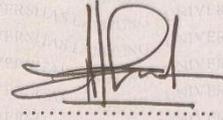


Dwi Sakethi, S.Si., M.Kom
NIP. 19680611 199802 1 001

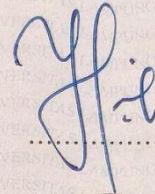
MENGESAHKAN

1. Tim Penguji

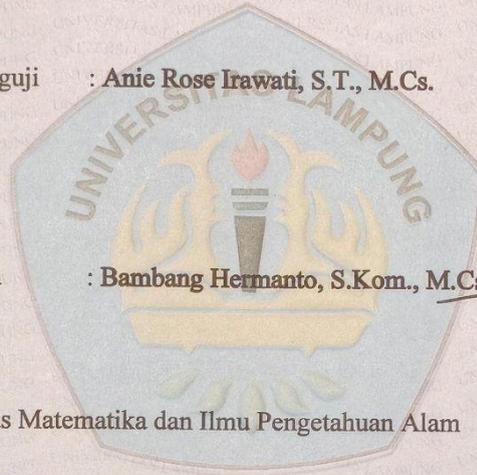
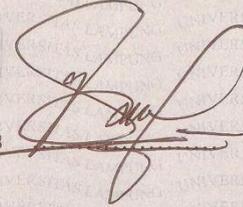
Ketua Penguji : Dr. rer. nat. Akmal Junaidi, M.Sc.



Sekretaris Penguji : Anie Rose Irawati, S.T., M.Cs.



Penguji Utama : Bambang Hermanto, S.Kom., M.Cs



2. Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam



Dr. Eng. Heri Satria, S.Si., M. Si.
NIP. 19711001 200501 1 002

Tanggal Lulus Ujian Skripsi: 12 Desember 2024

PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Nur Setiowati

NPM : 2017051013

Menyatakan bahwa skripsi saya yang berjudul "**Implementasi *Software Product Quality* Berdasarkan Model ISO/IEC 25010 dalam Pengujian *Website* PT.Indonesia Teknologi Preneur (Studi Kasus: Disnaker Kabupaten XYZ)**" merupakan karya saya sendiri dan bukan karya orang lain. Semua tulisan yang tertuang di skripsi ini telah mengikuti kaidah penulisan karya ilmiah Universitas Lampung. Apabila di kemudian hari terbukti skripsi saya merupakan hasil penjiplakan atau dibuat orang lain, maka saya bersedia menerima sanksi berupa pencabutan gelar yang telah saya terima.

Bandar Lampung, 12 Desember 2024



Nur Setiowati
NPM. 2017051013

RIWAYAT HIDUP



Lahir di Kabupaten Tanggamus, 19 Januari 2002. Anak kedua dari empat bersaudara, yang menyelesaikan pendidikan di SDN 01 Kebumen Kabupaten Tanggamus pada tahun 2014, kemudian menyelesaikan pendidikan menengah pertama di SMPN 02 Sumberejo pada tahun 2017, dan lulus dari pendidikan menengah atas di SMAN 1 Sumberejo pada tahun 2020.

Pada tahun 2020, terdaftar sebagai mahasiswa Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung melalui jalur SNMPTN. Kegiatan-kegiatan yang dilakukan selama menjadi mahasiswa yaitu sebagai berikut.

1. Menjadi Asisten Dosen mata kuliah Logika di Jurusan Ilmu Komputer pada tahun ajaran Ganjil 2021/2022
2. Menjadi Asisten Dosen mata kuliah Matematika Diskrit di Jurusan Ilmu Komputer pada tahun ajaran Genap 2021/2022.

3. Menjadi Anggota Bidang Keilmuan Himpunan Mahasiswa Jurusan Ilmu Komputer periode 2021/2022.
4. Menjadi Sekretaris Pelaksana HINCAR TIK HIMAKOM Jurusan Ilmu Komputer periode 2022.
5. Melaksanakan Program *Short Course* Desain Interaksi Untuk UI/UX Designer Pemula Kredensial Mikro Mahasiswa Indonesia (KMMI) di jurusan Ilmu Komputer Fakultas MIPA Unila periode 2021/2022.
6. Melaksanakan Kerja Praktek pada bulan Desember – Januari periode 2022/2023 di Badan Pengawasan Keuangan dan Pembangunan Provinsi Lampung.
7. Melaksanakan Studi Independen – Kampus Merdeka (MBKM) pada bulan Februari – Juni periode 2022/2023 Program *Be a Top Search Quality Engineer* di Alterra Academy.
8. Melaksanakan Kuliah Kerja Nyata Kota Bandar Lampung Pada tahun 2022/2023.

MOTO

“Jika kalian berbuat baik, sesungguhnya kalian berbuat baik bagi diri kalian sendiri”

(QS. Al-Isra:7)

“Barangsiapa yang memudahkan kesulitan seorang mu'min dari berbagai kesulitan-kesulitan dunia, Allah akan memudahkan kesulitan-kesulitannya pada hari kiamat. Dan siapa yang memudahkan orang yang sedang dalam kesulitan niscaya akan Allah memudahkan baginya di dunia dan akhirat”

(HR. Muslim)

“Jika hidup adalah seni dalam memilih, maka pilihlah kesulitan yang paling bisa dihadapi dan kebahagiaan yang ingin dicapai”

(Nur Setiowati)

“Jika tidak berani memperjuangkan mimpimu, tidak pantas adalah kata yang lebih layak dari penyesalan”

(Nur Setiowati)

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

PERSEMBAHAN

Alhamdulillahillobbilamin

Puji dan syukur tercurahkan kepada Allah Subhanahu Wa Ta'ala atas segala Rahmat dan Karunia-Nya sehingga saya dapat menyelesaikan skripsi ini. Shalawat serta salam selalu tercurahkan Kepada Nabi Muhammad SAW.

Kupersembahkan karya ini kepada:

Keluarga Tercinta

Ayah, Ibu, Kakak, dan Adik

Yang senantiasa memberikan yang terbaik dan melantunkan do'a yang selalu Menyertaiku. Kuucapkan pula terimakasih sebesar-besarnya kepada Ayah dan Ibu karena telah mendidik dan membesarkanku dengan cara yang dipenuhi kasih sayang, dukungan, dan pengorbanan. Kupersembahkan semua ini untuk kalian, Tanpa kalian, semua ini tidak akan mungkin terwujud. Setiap langkah yang kuambil adalah hasil dari cinta dan kerja keras kalian.

Seluruh Keluarga Besar Ilmu Komputer 2020

Yang selalu memberikan semangat dan dukungan.

Almater Tercinta, Universitas Lampung dan Jurusan Ilmu Komputer

Tempat menimba ilmu, untuk menjadi bekal hidup dunia dan akhirat.

SANWACANA

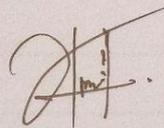
Alhamdulillah Rabbil ‘Alamin. Segala puji dan syukur penulis panjatkan kehadirat Allah Subhanahu Wa Ta’ala yang telah memberikan nikmat dan hidayah-Nya sehingga penulis dapat menyelesaikan skripsi ini. Sholawat serta salam penulis sanjungkan kepada Baginda Nabi Muhammad Shallallahu ‘Alaihi Wasallam yang penulis harapkan syafaatnya di hari akhir kelak.

Skripsi yang berjudul “Implementasi *Software Product Quality* Berdasarkan Model ISO/IEC 25010 dalam Pengujian *Website* PT. Indonesia Teknologi Preneur (Studi Kasus: Disnaker Kabupaten XYZ)” ini disusun sebagai salah satu syarat untuk memperoleh gelar sarjana jurusan Ilmu Komputer Universitas Lampung. Selesaiannya Skripsi ini tidak terlepas dari bantuan, bimbingan dan dukungan dari berbagai pihak. Oleh karena itu, ucapan terima kasih ditujukan kepada:

1. Allah Subhanahu Wa Ta’ala yang telah memberikan nikmat kesehatan dan kemampuan untuk menyelesaikan skripsi ini.
2. Ayahanda dan Ibunda tercinta, kakak, Adik serta Keluarga besar yang selalu memberikan do'a dan dukungan.
3. Bapak Dr. Eng. Heri Satria, S.Si., M.Si. selaku Dekan FMIPA Universitas Lampung.
4. Bapak Dwi Sakethi, S.Si., M.Kom. selaku ketua Jurusan Ilmu Komputer FMIPA Universitas Lampung.
5. Ibu Anie Rose Irawati S.T., M.Cs. selaku Sekretaris Jurusan Ilmu Komputer FMIPA Universitas Lampung.
6. Bapak Febi Eka Febriansyah, M.T. sebagai Dosen Pembimbing Akademik yang telah memberikan arahan, ide, motivasi, dan dukungan akademik penulis

7. Bapak Dr. rer. nat. Akmal Junaidi, M.Sc. sebagai Dosen Pembimbing Utama yang telah meluangkan tenaga, waktu dan pikirannya dalam memberikan arahan, ide, motivasi, kritik serta saran kepada penulis sehingga dapat menyelesaikan skripsi ini dengan baik.
8. Ibu Anie Rose Irawati, S.T., M.Cs. sebagai Dosen Pembimbing Pembantu yang juga senantiasa meluangkan tenaga, waktu dan pikirannya dalam memberikan arahan, ide, motivasi, kritik serta saran kepada penulis sehingga dapat menyelesaikan skripsi ini dengan baik.
9. Bapak Bambang Hermanto, S.Kom., M.Cs. sebagai Dosen Pembahas yang telah memberikan masukan yang bermanfaat dalam perbaikan skripsi ini dan mendukung peningkatan akademik penulis.
10. Ibu Ade Nora Maela, Bang Zainuddin dan Mas Nofal yang telah membantu segala urusan administrasi penulis di Jurusan Ilmu Komputer.
11. Bapak dan Ibu Dosen Jurusan Ilmu Komputer FMIPA Universitas Lampung yang telah memberikan ilmu dan pengalaman dalam hidup untuk menjadi lebih baik.
12. Keluarga Besar Ilmu Komputer 2020 yang tidak bisa penulis sebutkan satu persatu.

Bandar Lampung, 12 Desember 2024



Nur Setiowati
NPM. 2017051013

DAFTAR ISI

	Halaman
DAFTAR ISI.....	2
DAFTAR TABEL.....	4
DAFTAR GAMBAR.....	5
I. PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	5
1.3 Tujuan.....	6
1.4 Manfaat.....	6
1.5 Batasan Masalah.....	6
II. TINJAUAN PUSTAKA.....	7
2.1 Profile PT Indonesia Teknologi Preneur (Indotechpren).....	7
2.2 Penelitian Terdahulu.....	8
2.3 Landasan Teori.....	12
2.3.1 Pengujian.....	12
2.3.2 Pengujian Perangkat Lunak.....	12
2.3.3 Kualitas Perangkat Lunak.....	13
2.3.4 <i>Web Browser</i>	13
2.3.5 <i>Website</i>	14
2.3.6 <i>Blackbox Testing</i>	14
2.3.7 <i>Software Testing Life Cycle (STLC)</i>	17

2.3.8 Disnaker.....	20
2.3.9 ISO/IEC 25010.....	21
2.3.10 Karakteristik <i>Usability</i> ISO/IEC 25010.....	27
2.3.11 Karakteristik <i>Portability</i> ISO/IEC 25010.....	30
2.3.12 Karakteristik <i>Fungsional Suitability</i> ISO/IEC 25010.....	34
2.3.13 Karakteristik <i>Performance Efficiency</i> ISO/IEC 25010.....	36
2.3.14 Alat Pengujian Perangkat Lunak (<i>Software Testing Tools</i>).....	39
2.3.15 Dimensi Kualitas <i>Website</i> Pressman.....	43
2.3.16 <i>Automation Testing</i>	44
2.3.17 Test Template.....	44
2.3.18 <i>Bug</i>	45
2.3.19 <i>User Acceptance Testing</i> (UAT).....	46
III. METODE PENELITIAN.....	47
3.1 Sumber Data.....	47
3.2 Waktu dan Tempat Penelitian.....	48
3.3 Alat Penelitian.....	49
3.4 Alur dan Tahapan Penelitian.....	50
3.4.1 <i>Requirement Analysis</i>	51
3.4.2 <i>Test Planning</i>	63
3.4.3 <i>Test Case Development</i>	70
3.4.4 <i>Environment Setup</i>	71
3.4.5 <i>Test Execution</i>	71
3.4.6 <i>Test Cycle Closure</i>	72
IV. HASIL DAN PEMBAHASAN.....	73
4.1 <i>Test Case Development</i>	73
4.1.1 <i>Test Case Development</i> Karakteristik <i>Usability</i>	73
4.1.2 <i>Test Case Development</i> Karakteristik <i>Portability</i>	74
4.1.3 <i>Test Case Development</i> Karakteristik <i>Fungsional Suitability</i>	8

4.1.4 <i>Test Case Development</i> Karakteristik <i>Performance Efficiency</i>	89
4.2 <i>Environment Setup</i>	89
4.2.1 <i>Environment Setup</i> Karakteristik <i>Usability</i>	89
4.2.2 <i>Environment Setup</i> Karakteristik <i>Portability</i>	93
4.2.3 <i>Environment Setup</i> Karakteristik <i>Fungsional Suitability</i>	95
4.2.4 <i>Environment Setup</i> Karakteristik <i>Performance Efficiency</i>	97
4.3 <i>Test Execution</i>	99
4.3.1 <i>Test Execution</i> Karakteristik <i>Usability</i>	100
4.3.2 <i>Test Execution</i> Karakteristik <i>Portability</i>	101
4.3.3 <i>Test Execution</i> Karakteristik <i>Fungsional Suitability</i>	112
4.3.4 <i>Test Execution</i> Karakteristik <i>Performance Efficiency</i>	116
4.3 <i>Test Cycle Closure</i>	116
4.4.1 <i>Test Cycle Closure</i> Karakteristik <i>Usability</i>	116
4.4.2 <i>Test Cycle Closure</i> Karakteristik <i>Portability</i>	116
4.4.3 <i>Test Cycle Closure</i> Karakteristik <i>Fungsional Suitability</i>	116
4.4.4 <i>Test Cycle Closure</i> Karakteristik <i>Performance Efficiency</i>	117
V. KESIMPULAN DAN SARAN	118
5.1 Kesimpulan.....	118
5.2 Saran.....	119
DAFTAR PUSTAKA	120

DAFTAR TABEL

Tabel	Halaman
1. Penelitian Terdahulu.....	9
2. Nilai dan <i>Grade Usability Testing Tools Maze</i>	42
3. Model <i>Testcase</i>	45
4. Kebutuhan Fungsional Peserta.....	60
5. Kebutuhan Non-Fungsional Website Disnaker.....	62
6. Skor Likert.....	63
7. Indeks Kepuasan Pengguna.....	64
8. Test Plan Secara Umum.....	66
9. Prosedur Pengujian.....	67
10. Test Plan Karakteristik <i>Usability</i>	68
11. Karakteristik Partisipan <i>Usability Testing</i>	68
12. Nilai dan <i>Grade Portability Testing</i>	71
13. <i>Test Case Portability Testing Windows</i>	77
14. <i>Test Case Portability Testing MacOS</i>	79
15. Test Case Platform Mobile.....	82
15. Test Case Karakteristik <i>Fungsional Suitability</i>	84
16. Pernyataan <i>User Acceptance Testing (UAT)</i>	92
17. Test Case Karakteristik <i>Performance Efficiency</i>	93
18. <i>Test Execution Portability Testing Sistem Operasi Windows</i>	106
19. <i>Test Execution Portability Testing Sistem Operasi MacOS</i>	108
20. <i>Test Execution Portability Testing Sistem Platform Mobile</i>	111
21. <i>Test Execution</i>	114
22. Dokumentasi Kesalahan Sistem.....	115
23. Hasil <i>User Acceptance Testing (UAT)</i>	123
24. Metrik <i>Performance Testing</i>	124

DAFTAR GAMBAR

Gambar	Halaman
1. Struktur Organisasi PT Indotechpren.....	8
2. Tampilan Utama Website Disnaker.....	20
3. Dimensi <i>Quality in Use</i> ISO/IEC 25010.....	21
4. Dimensi <i>Software Product Quality</i> ISO/IEC 25010.....	23
5. Waktu Penelitian.....	50
6. <i>Software Testing Life Cycle</i> (STLC).....	51
7. Kegiatan <i>Meeting Online</i>	59
8. Tampilan Lembar Kerja Baru pada Maze.....	94
9. Tampilan “ <i>Add Block</i> ” pada Maze.....	94
10. Tampilan Detail Rancangan Misi/Tugas pada Maze.....	95
11. Tampilan “ <i>Add Block</i> ” pada Useberry.....	96
12. Tampilan Detail Rancangan Misi/Tugas pada Useberry.....	96
13. Fitur <i>Live</i> pada Browserstack.....	97
14. <i>Dashboard</i> Browserstack <i>Featured Live</i>	98
16. Konfigurasi Selenium IDE.....	100
17. <i>Test Case</i> pada Selenium IDE.....	100
18. <i>Running Test Case</i> pada Selenium IDE.....	100
19. <i>Filter Pattern</i> pada Blazemeter.....	102
20. <i>Start Recording</i> pada Blazemeter.....	102
21. <i>Add Listener</i> pada JMeter.....	103
22. <i>Setting Thread Group</i> pada JMeter.....	104
23. Hasil <i>Usability Testing Tools</i> Maze.....	105
24. Hasil <i>Usability Testing Tools</i> Useberry.....	106
25. Eksekusi <i>Test Case Portability Testing Tools</i> Browserstack.....	106
26. Tampilan Awal <i>Portability Testing</i> Windows.....	107

27. Hasil <i>Portability Testing</i> OS Windows Tools Browserstack.....	108
28. Tampilan Awal <i>Portability Testing</i> macOS.....	109
29. Hasil <i>Portability Testing</i> macOS Tools Browserstack.....	110
30. Tampilan Awal <i>Portability Testing</i> macOS.....	110
31. Hasil <i>Portability Testing FAIL</i> macOS Tools Browserstack.....	111
32. Hasil <i>Portability Testing Platform Mobile Tools</i> Browserstack.....	112
33. Inspeksi Elemen.....	117
34. Inspeksi Elemen Pendaftaran Pelatihan Dan Sertifikasi.....	117
35. Dashboard Admin Menu Sertifikasi.....	118
36. Gambar Lebih Besar Dari 4,52 MB.....	119
37. Inspeksi Elemen <i>Upload File</i> Lebih Dari 2 MB.....	119
38. <i>Upload File</i> Dengan Format .webp.....	120
39. <i>Inspect Element Upload File</i> Dengan Format .webp.....	120
40. Hasil Uji Validitas.....	122
41. Hasil Uji Reliabilitas.....	123
42. Langkah <i>Test Execution Tools</i> JMeter.....	126

I. PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi saat ini bagaikan dua mata pisau yang perlu mendapatkan perhatian khusus dari semua elemen. Pengguna internet di Indonesia hingga saat ini telah mencapai 221 juta orang (<https://www.apjii.or.id/>). Tuntutan perkembangan teknologi di Indonesia semakin memaksa perusahaan di berbagai sektor industri untuk beradaptasi melalui digitalisasi dalam meningkatkan daya saing dan efisiensi operasional. Dalam era ini, banyak perusahaan mengambil langkah dengan memanfaatkan teknologi digital, salah satunya melalui penggunaan sistem informasi berbasis *website* agar perusahaan dapat lebih mudah menjangkau konsumen, memasarkan produk atau layanan secara *online*, dan meningkatkan visibilitas mereka di dunia maya. Selain itu, *website* juga menjadi sarana efektif untuk berkomunikasi dengan pelanggan, menyediakan informasi produk atau layanan, dan mendukung aktivitas bisnis secara keseluruhan. Penerapan sistem informasi berbasis *website* tidak hanya berlaku di sektor bisnis, tetapi juga merambah ke sektor pendidikan, pemerintahan, kesehatan, dan berbagai sektor lainnya turut memanfaatkan teknologi digital guna meningkatkan kualitas layanan dan efisiensi proses kerja. Dengan demikian, digitalisasi melalui *website* bukan hanya menjadi tren, tetapi juga kebutuhan strategis bagi perusahaan dan entitas lainnya di Indonesia. Hal ini menunjukkan bahwa perkembangan teknologi memiliki dampak besar pada transformasi berbagai sektor dalam masyarakat, mengubah cara mereka beroperasi dan berinteraksi dengan lingkungan sekitar.

Website atau situs merujuk pada sejumlah halaman yang menampilkan berbagai jenis informasi seperti teks, gambar diam atau bergerak, animasi, suara, video, atau kombinasi dari semua itu. Halaman-halaman ini bisa bersifat statis atau dinamis, dan mereka saling terhubung membentuk struktur yang terorganisir, dihubungkan melalui jaringan-jaringan halaman (Riyadi dkk, 2012). *Website* memiliki banyak manfaat diantaranya adalah pembangun branding bisnis, media promosi, alat pelayanan konsumen, tempat kritik dan saran serta alat untuk melakukan interaksi (Huda, 2020). Hal tersebut membuka peluang besar bagi perusahaan di bidang Teknologi Informasi (TI) untuk menyediakan berbagai layanan digital yang dapat mendukung kebutuhan perusahaan dan instansi dalam era digital ini. Salah satu perusahaan yang beroperasi di Lampung dan menawarkan layanan-layanan TI tersebut adalah PT Indonesia Teknologi Preneur.

PT Indonesia Teknologi Preneur (Indotechpren) merupakan salah satu *Startup* di provinsi Lampung yang didirikan oleh Putra Pribowo, S.Kom selaku CEO pada 1 Januari tahun 2021. PT Indonesia Teknologi Preneur bergerak dalam bidang jasa dengan layanan diantaranya *Computer Network, Software, Cyber Security, CCTV*, dan Pelatihan. Dalam bidang *Software* PT Indonesia Teknologi Preneur menyediakan jasa untuk membangun *website* dan aplikasi (<https://www.indotechpren.org/>, 2021). Dengan spesialisasi dalam penyediaan solusi digital, PT Indonesia Teknologi Preneur dapat membantu perusahaan dan instansi di wilayah sekitar untuk beradaptasi dengan perubahan zaman yang serba digital. Melalui layanan-layanan mereka, seperti pengembangan *website*, pembuatan aplikasi khusus, dan integrasi sistem, PT Indonesia Teknologi Preneur dapat memainkan peran kunci dalam mendukung transformasi digital di berbagai sektor. Dengan berkembangnya permintaan akan layanan-layanan TI ini, perusahaan seperti PT Indonesia Teknologi Preneur dapat turut serta dalam memajukan perekonomian lokal dan memfasilitasi inovasi di tingkat regional. Hal ini menciptakan lingkungan di mana perusahaan, lembaga pemerintah, dan organisasi non-profit dapat meningkatkan efisiensi, meningkatkan daya saing, dan memberikan pelayanan yang lebih baik kepada masyarakat. Dengan demikian,

integrasi teknologi digital bukan hanya menjadi kebutuhan, tetapi juga peluang bagi pengembangan bisnis dan kemajuan di berbagai sektor.

Terkait dengan pembangunan perangkat lunak yang ditawarkan oleh PT Indonesia Teknologi Preneur, penting untuk memastikan bahwa perangkat lunak tersebut memenuhi standar kualitas dan harapan pelanggan. Dalam konteks ini, proses pengujian atau *testing* menjadi sangat penting. *Testing* merupakan langkah eksekusi suatu program atau sistem dengan tujuan untuk mengidentifikasi dan mengatasi potensi kecacatan atau *bug* dan menyempurnakan kualitas perangkat lunak agar sesuai dengan kebutuhan dan spesifikasi yang telah ditetapkan. Dengan mengidentifikasi dan memperbaiki masalah potensial, perangkat lunak dapat berfungsi dengan lebih baik, memberikan pengalaman pengguna yang lebih lancar, dan sesuai dengan harapan serta kebutuhan pengguna. Kepuasan pengguna menjadi faktor kunci dalam pengembangan perangkat lunak yang sukses. Oleh karena itu, tujuan akhir dari *testing* adalah untuk menyediakan produk perangkat lunak yang berkualitas tinggi dan memuaskan pengguna (Pressman, 2014). Tujuan lainnya dari pengujian perangkat lunak adalah menilai apakah tahap pengembangan perangkat lunak telah sesuai dengan metodologi yang digunakan, dan membuat dokumentasi hasil pengujian yang menginformasikan kesesuaian perangkat lunak yang diuji dengan spesifikasi yang telah ditentukan. Dengan menguji dan memastikan bahwa perangkat lunak beroperasi sesuai dengan yang diinginkan, perusahaan atau pengembang dapat meningkatkan kepuasan pelanggan, membangun reputasi yang baik, dan menjaga loyalitas pelanggan.

Dalam membangun perangkat lunak PT Indonesia Teknologi Preneur telah melakukan pengujian terhadap perangkat lunak sebelum dirilis dan diberikan kepada pengguna. Di perusahaan tersebut, pengujian menggunakan teknik *black box testing* telah dilakukan, akan tetapi belum secara maksimal dikarenakan belum sepenuhnya mempertimbangkan penggunaan teknik *blackbox testing* secara keseluruhan, hanya menggunakan beberapa saja dan kebanyakan hanya menguji partisi yang benar, sehingga kesalahan masih mungkin ditemukan. Dalam penelitian ini pengujian terhadap *website* Disnaker Akan dilakukan dengan

memaksimalkan penggunaan teknik pada metode *Blackbox Testing*. *Blackbox* merupakan pendekatan pengujian untuk mengetahui apakah semua fungsi perangkat lunak telah berjalan semestinya sesuai dengan kebutuhan fungsional yang telah didefinisikan (Pressman, 2010). Kasus uji ini bertujuan untuk menunjukkan fungsi perangkat lunak tentang cara beroperasinya. Teknik pengujian ini berfokus pada domain informasi dari perangkat lunak, yaitu melakukan kasus uji dengan mempartisi domain input dan output program (Jiang, 2012). Metode *blackbox* memungkinkan perancang perangkat lunak mendapatkan serangkaian kondisi input yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program.

Meskipun pengujian fungsional secara *black box* telah dilakukan, namun masih belum optimal karena dilakukan tanpa mempertimbangkan metode pengembangan *test case* seperti BVA (*Boundary Value Analysis*), *Decision Table*, *Equivalence Partitioning Technique* dan lain sebagainya yang dapat secara efektif menemukan adanya kesalahan pada sistem. Selain itu, dokumentasi pengujian sudah ada akan tetapi kurang lengkap. Beberapa pengujian lainnya, seperti pengujian performa sudah dilakukan sendiri oleh CEO perusahaan tersebut. Keputusan ini diambil karena CEO ingin terlibat langsung dalam proyek, sehingga pembuatan aplikasi dapat lebih optimal sesuai dengan kebutuhan pelanggan. Alasan lainnya adalah untuk menghindari penambahan biaya yang cukup tinggi jika pengujian dilakukan secara khusus oleh *software tester* dengan menambahkan anggota tim pengembang perangkat lunak. Selain itu, pengujian lainnya seperti *Usability Testing* dan *Portability Testing* belum dilakukan oleh PT Indonesia Teknologi Preneur.

Software dapat dikatakan sebagai suatu produk yang memiliki kualitas atau nilai (*value*) tertentu yang melekat padanya. Maksud dari *value* dalam konteks perangkat lunak adalah manfaat atau kualitas yang melekat pada perangkat lunak yang dapat memberikan kepuasan kepada penggunanya. *Value* ini mencerminkan sejauh mana perangkat lunak mampu memenuhi kebutuhan pengguna serta tujuan bisnis yang diinginkan. Adapun *value* yang dimaksud diantaranya adalah

fungsionalitas, reliabilitas, kemudahan penggunaan, efisiensi, portabilitas, keamanan dan pemeliharaan. Kualitas ini menjadi indikator penting dalam menentukan keberhasilan sebuah perangkat lunak, termasuk website yang merupakan salah satu jenis *software* yang banyak digunakan. Sebagai salah satu bentuk implementasi teknologi informasi, website berfungsi tidak hanya sebagai alat komunikasi, tetapi juga sebagai sarana untuk meningkatkan efisiensi operasional dan memberikan pengalaman pengguna yang optimal. Oleh karena itu, pengujian perangkat lunak seperti website sangat penting untuk memastikan bahwa kualitasnya sesuai dengan kebutuhan dan harapan pengguna. Hal ini menunjukkan bahwa terdapat aspek-aspek penting dalam pengujian perangkat lunak yang masih memerlukan perhatian lebih, terutama dalam memastikan kualitas sistem secara menyeluruh. Dengan mengamati berbagai proses yang terjadi selama pembangunan perangkat lunak, akhirnya timbul ketertarikan untuk melakukan penelitian yang menguji *website* Disnaker menggunakan ISO/IEC 25010 sebagai panduan dan pedoman pengujian perangkat lunak. Disnaker merupakan sebuah situs layanan terpadu ketenagakerjaan berbasis *online* meliputi pelayanan bidang ketenagakerjaan untuk masyarakat, perusahaan, dan kelembagaan milik salah satu kabupaten di provinsi Sumatera Selatan yang menyediakan informasi terkait dengan ketenagakerjaan, pelatihan dan sertifikasi, atau hal-hal terkait dengan bidang tersebut di wilayah tersebut yang mencakup informasi tentang kebijakan, program-program pemerintah, layanan-layanan yang tersedia.

Metode ISO/IEC 25010 dapat digunakan untuk mengevaluasi kualitas sistem perangkat secara spesifik berdasarkan dua dimensi umum, yaitu *quality in use* dan *product quality*. Pada *quality in use*, terdapat beberapa karakteristik relatif yang ditinjau dari perspektif *user*, antara lain *effectiveness*, *satisfaction*, *context coverage* dan *efficiency*. Sedangkan pada dimensi *product quality*, di mana prosesnya mengacu pada karakteristik intrinsik dari sebuah produk perangkat lunak, memiliki beberapa elemen yang antara lain meliputi *functional suitability*, *reliability*, *performance efficiency*, *Usability*, *Security*, *compatibility*, *maintainability* dan *Portability*. Adapun dimensi yang digunakan untuk menguji

website Disnaker dalam penelitian ini diantaranya adalah *Software Product Quality Model* dengan karakteristik *Functional Suitability*, *Performance Efficiency*, *Usability*, dan *Portability*. Pengujian akan dilakukan secara otomatis menggunakan *tools* khusus untuk masing-masing karakteristik yang akan diuji seperti Selenium, Katalon Recorder, Maze, Useberry, JMeter, Blazemeter dan Browserstack. Adapun pengujian fungsional akan dilakukan menggunakan metode *blackbox testing* dengan dokumentasi skenario dan kasus uji yang detail dan terperinci. Hasil dari pemaparan tersebut adalah keputusan untuk melakukan penelitian dalam bidang pengujian perangkat lunak dengan judul “Implementasi *Software Product Quality* Berdasarkan Model ISO/IEC 25010 Dalam Pengujian *Website* PT. Indonesia Teknologi Preneur (Studi Kasus: Disnaker Kabupaten XYZ)“.

1.2 Rumusan Masalah

Adapun rumusan masalah yang dapat ditarik dari latar belakang yang telah dipaparkan adalah “Bagaimana Implementasi *Software Product Quality* Berdasarkan Model ISO/IEC 25010 dalam Pengujian *Website* PT. Indonesia Teknologi Preneur (Studi Kasus: Disnaker Kabupaten XYZ)?”.

1.3 Tujuan

Adapun tujuan dari penelitian yang dilakukan adalah mengimplementasikan *Software Product Quality* Berdasarkan Model ISO/IEC 25010 dalam Pengujian *Website* Disnaker Kabupaten XYZ yang dibuat oleh PT Indonesia Teknologi Preneur (Indotechpren).

1.4 Manfaat

Adapun manfaat dari penelitian yang dilakukan adalah sebagai berikut.

1. Membantu melakukan pengujian menggunakan *Software Product Quality Model* ISO/IEC 25010 pada *website* Disnaker yang dibuat oleh PT Indonesia Teknologi Preneur.
2. Mengetahui hasil pengujian berdasarkan karakteristik yang diuji menggunakan *Software Product Quality Model* ISO/IEC 25010.

3. Adanya dokumentasi pengujian yang terperinci, lengkap, dan rapi. Sehingga memudahkan pengembang dalam menemukan kesalahan dalam sistem yang serupa dan membantu CEO serta pemilik perangkat lunak memahami perangkat lunak dengan lebih baik.

1.5 Batasan Masalah

Adapun batasan masalah dari penelitian yang dilakukan adalah sebagai berikut.

1. Pengujian dilakukan pada *website* Disnaker Kabupaten XYZ milik PT Indonesia Teknologi Preneur dan terbatas pada fitur yang dipilih di *role* peserta.
2. Penelitian ini terbatas pada karakteristik ISO/IEC 25010 yang diuji diantaranya yaitu *Functional Suitability*, *Performance Efficiency*, *Usability* dan *Portability*.
3. Penelitian ini terbatas pada alat pengujian yang digunakan serta hasil akhir dari penelitian ini adalah angka atau *grade* yang merupakan nilai dari karakteristik ISO/IEC 25010 yang diuji.

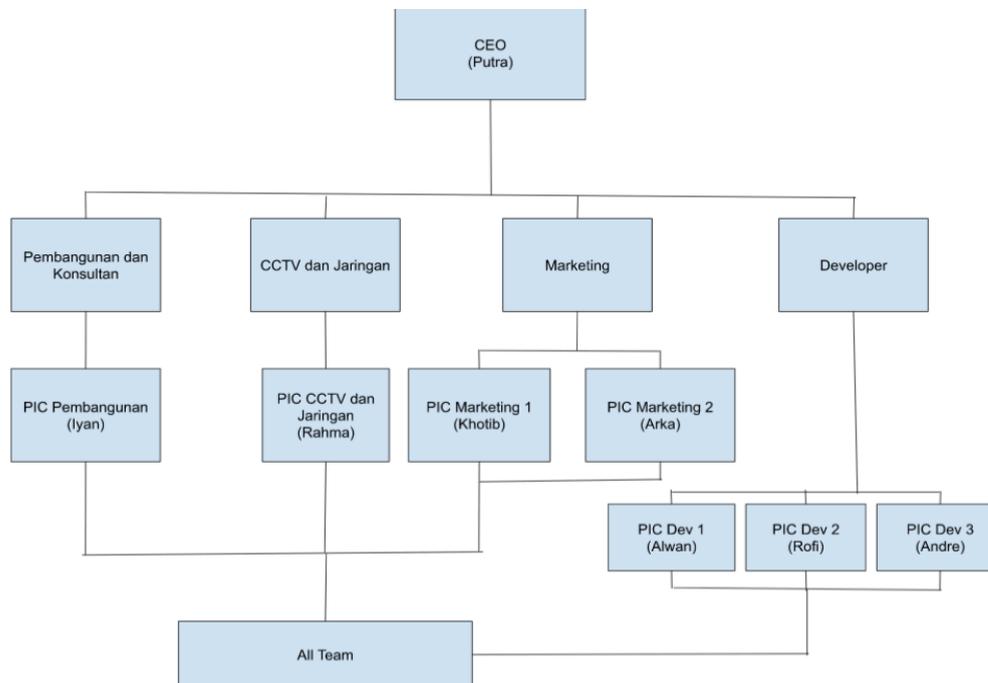
II. TINJAUAN PUSTAKA

2.1 Profile PT Indonesia Teknologi Preneur (Indotechpren)

PT Indonesia Teknologi Preneur (Indotechpren) merupakan salah satu *Startup* di provinsi Lampung yang didirikan oleh Putra Pribowo, S.Kom selaku CEO pada 1 Januari tahun 2021. PT Indotechpren bergerak dalam bidang teknologi yang menawarkan jasa dengan layanan diantaranya *Computer Network, Software, Cyber Security, CCTV, dan Pelatihan*. Dalam bidang *Software* PT Indotechpren menyediakan jasa untuk membangun *website* dan aplikasi. Adapun visi dari PT Indotechpren adalah “Menjadikan perusahaan teknologi nomor 1 dalam bidang *Computer & Network, Software, Cyber Security, CCTV, Pelatihan* dan membantu dalam membangun SDM di Lokal, Nasional dan, Internasional.”, dengan misi antara lain menggunakan teknologi terbaru dalam sistem perusahaan, memberikan pelatihan teknologi di tingkat Lokal, Nasional, dan Internasional, meningkatkan kemampuan sumber daya manusia yang handal, membangun kerjasama berbagai pihak, dan memberikan pelayanan maksimal kepada *customer*.

PT Indotechpren telah menjalin kerja sama dengan BKIPM Lampung, Inkubator Universitas Lampung, Jurusan Ilmu Komputer Universitas Lampung, Reana, projects.co.id, Capsuleinn, PT Arti Media Nusantara, Way Kambas Lampung, Pondok Ahmad Ghozali dan Hobikoe. Aplikasi Appikceria berbasis *website* dan mobile milik BKIPM Lampung, Aplikasi SPD berbasis *website* milik BPKP Lampung, Aplikasi Stok Gudang berbasis *website* milik PT Dinasty Han Djaya, Greenmetrick Unila, Aplikasi Pameran Virtual, Aplikasi Sensor tempat dengan *network* WIFI dan Aplikasi Siakadu milik Universitas Lampung milik Universitas

Lampung merupakan beberapa project yang telah diselesaikan oleh PT Indotechpren (<https://www.indotechpren.org/>, 2021).



Gambar 1. Struktur Organisasi PT Indotechpren.

Gambar 1 merupakan ilustrasi dari struktur organisasi dari PT Indotechpren dipimpin oleh CEO atas nama Putra. PT Indotechpren terletak di Gg. Rukun Nomor 1, Bumi, Kecamatan Kedamaian, Kota Bandar Lampung. PT Indotechpren terdiri dari beberapa bidang antara lain Pembangunan dan konsultan dengan PIC Iyan, CCTV dan Jaringan dengan PIC Rahma, Marketing dengan PIC 1 Khotib dan PIC 2 Arka dan *Developer* dengan PIC 1 Alwaan, PIC 2 Rofi dan PIC 3 Andre.

2.2 Penelitian Terdahulu

Penelitian terdahulu merupakan kajian dan analisis literatur yang sebelumnya telah dilakukan oleh peneliti atau akademisi lain yang relevan dengan topik penelitian yang akan diteliti sehingga dapat dijadikan referensi dalam melakukan penelitian. Penelitian terdahulu ini juga digunakan untuk menemukan pembaruan dari penelitian yang akan dilakukan. Berikut ini disajikan beberapa penelitian terdahulu yang digunakan dalam penelitian.

Tabel 1. Penelitian Terdahulu.

Peneliti & tahun	Judul	Hasil Penelitian	Persamaan	Perbedaan
Ismail Tang-kudung /2019	Evaluasi <i>Website</i> Menggunakan Metode ISO/IEC 25010.	Hasil dari evaluasi dengan metode ISO/IEC 25010 pada penelitian ini adalah aspek <i>functional suitability</i> memenuhi syarat dengan nilai $x = 1$ (baik), aspek <i>performance efficiency</i> dengan skor rata-rata 45.18, <i>grade F</i> untuk <i>page speed</i> dan <i>Yslow</i> skor rata-rata 62.53, <i>grade D</i> serta rata-rata waktu respon 17.64 detik (butuh perbaikan), aspek <i>portability</i> menggunakan 3 <i>browser</i> dan hasilnya website dapat berjalan tanpa <i>error</i> (baik), dan aspek <i>usability</i> memiliki nilai SUS sebesar 57.38 dengan <i>grade scale D</i> .	<ul style="list-style-type: none"> - Menggunakan 4 karakteristik ISO/IEC 25010 yaitu <i>Fungsional suitability, usability, portability</i> dan <i>performance efficiency</i>. - Hasil pengujian setiap karakteristik disajikan berupa nilai dan <i>grade</i>. 	<ul style="list-style-type: none"> - Pengujian karakteristik dilakukan secara manual dan otomatis, sedangkan pada penelitian yang akan dilakukan semua karakteristik diuji secara otomatis menggunakan <i>tools</i>. - Perbedaan pada <i>tools</i> yang digunakan untuk menguji setiap karakteristik.
Shasa Intyana /2019	Implementasi ISO/IEC 25010 Untuk Meng-evaluasi <i>Website</i> SMA Negeri di Kabupaten Pringsewu	Karakteristik <i>usability</i> berpengaruh signifikan terhadap kepuasan pengguna, <i>compability</i> memiliki pengaruh namun tidak signifikan, <i>performance efficiency</i> memiliki pengaruh terhadap kepuasan pengguna meskipun tidak signifikan.	<ul style="list-style-type: none"> - Menggunakan ISO/IEC 25010. - Menguji karakteristik <i>fungsional suitability, performance efficiency, usability</i> dan <i>portability</i>. 	<ul style="list-style-type: none"> - Pengujian tersebut berfokus pada semua karakteristik yang berkaitan dengan kepuasan pengguna, sedangkan pada penelitian yang akan dilakukan fokus terhadap

Tabel 1. (lanjutan)

Peneliti & tahun	Judul	Hasil Penelitian	Persamaan	Perbedaan
		Namun untuk pengaruh signifikan antara portability dengan kepuasan pengguna hanya pada SMA Negeri 2 Gadingrejo dan SMA Negeri 1 Pagelaran.		hasil dari masing - masing karakteristik yang diuji. - Perbedaan pada tools yang digunakan untuk menguji setiap karakteristik.
Bayu Rizaldi Putra /2021	Peng-aplikasian ISO/IEC 25010 untuk meng-evaluasi website SMKN 1 Palang karaya	<p>Pada aspek <i>functional suitability</i>, sistem telah memenuhi standar karena fungsi berjalan 100%. Pada aspek <i>performance efficiency</i>, sistem telah memenuhi standar dengan rata-rata waktu untuk memuat halaman 5,2 detik dan pengukuran <i>performa PageSpeed</i> sebesar 32% (<i>grade D</i>), YSlow sebesar 67% (<i>grade D</i>). Pada aspek <i>usability</i>, sistem telah memenuhi standar dengan tingkat persetujuan pengguna sebesar 76,97% yang dikategorikan layak.</p> <p>Pada aspek <i>reliability</i>, sistem telah memenuhi standar kriteria dengan hasil <i>session, hits</i> dan <i>pages</i> sebesar 100%. Pada aspek <i>portability</i>, sistem telah memenuhi standar kriteria karena berhasil dijalankan pada 5 jenis browser desktop yang diujicobakan tanpa terjadi</p>	<p>- Menggunakan ISO/IEC 25010.</p> <p>- Menguji karakteristik <i>funksional suitability, performance efficiency, usability</i> dan <i>portability</i>.</p>	<p>- Penelitian tersebut menguji karakteristik <i>funksional suitability, performance efficiency, usability, reliability</i> dan <i>portability</i>. Sedangkan pada penelitian ini karakteristik yang diuji adalah <i>funksional suitability, performance efficiency, usability</i> dan <i>portability</i>.</p> <p>- Penelitian tersebut berfokus pada kepuasan pengguna berdasarkan lima karakter ISO/IEC yang diuji. Sedangkan pada penelitian ini berfokus pada nilai masing - masing karakteristik ISO/IEC yang diuji dengan membandingkan</p>

Tabel 1. (lanjutan)

Peneliti & tahun	Judul	Hasil Penelitian	Persamaan	Perbedaan
		kesalahan. Website SMKN 1 Palangkaraya, layak dan sesuai untuk digunakan, sesuai dengan standar model ISO/IEC 25010, walaupun dari aspek performance efficiency untuk waktu standar rata-rata sudah memenuhi namun dari pengukuran performa PageSpeed dan YSlow masih kurang yaitu sebesar 32% (grade D) dan 67% (grade D).		<p>hasil dari kedua tools yang digunakan Pada penelitian tersebut pengujian pada setiap karakteristik dilakukan secara manual dan otomatis (tools). Sedangkan pada penelitian ini semua dilakukan secara otomatis menggunakan <i>tools</i>.</p> <p>- Lokasi penelitian yang berbeda yaitu pada penelitian tersebut dilakukan di SMKN 1 Palangkaraya, sedangkan pada penelitian ini dilakukan di PT Indotechpren</p>

2.3 Landasan Teori

Berikut ini merupakan berbagai teori yang digunakan dalam penelitian.

2.3.1 Pengujian

Pengujian adalah rangkaian kegiatan yang dipersiapkan dan dijalankan secara teratur guna menilai atau mengevaluasi ketepatan yang diinginkan. Proses pengujian terdiri dari sejumlah langkah yang dapat membantu dalam merancang kasus uji yang sesuai dan terperinci (Rosa & Shalahuddin, 2014).

2.3.2 Pengujian Perangkat Lunak

Pengujian perangkat lunak adalah langkah eksekusi sistem perangkat lunak yang bertujuan untuk mendeteksi kesalahan atau kecacatan dalam perangkat lunak tersebut (Myers, 1979). Pada dasarnya, perangkat lunak yang memiliki kualitas yang baik adalah perangkat lunak yang tidak memiliki kesalahan atau kecacatan secara objektif. Agar dapat mencapai hasil yang objektif, perangkat lunak harus melewati suatu proses pengujian yang terstruktur, terencana, dan terdokumentasi dengan baik. Pengujian yang dilakukan dengan baik memiliki kemungkinan yang tinggi dalam menemukan kesalahan (Pressman, 2014). Kecacatan secara objektif merujuk pada ketidaksempurnaan atau kekurangan dalam perangkat lunak yang dapat diukur atau diidentifikasi secara jelas dan tidak dapat disengketakan.

Dalam konteks pengujian perangkat lunak, kecacatan secara objektif dapat berupa kesalahan dalam fungsi-fungsi yang diharapkan, kinerja yang tidak memenuhi standar yang ditetapkan, atau pelanggaran terhadap persyaratan teknis. Singkatnya pengujian perangkat lunak adalah sebuah metode untuk mengetahui apakah software yang dibuat, berfungsi dengan baik dan benar. Tanpa adanya pengujian perangkat lunak kita tidak dapat mengetahui apakah sebuah *software* sudah memenuhi kriteria yang dibutuhkan untuk pengguna. Tahap pengujian merupakan hal yang sangat penting dalam pengembangan perangkat lunak dan dapat menentukan kualitas dari perangkat lunak itu sendiri. Perangkat lunak yang telah diuji sebelumnya cenderung memiliki kemungkinan kesalahan yang lebih kecil, sehingga dapat mengurangi risiko kegagalan ketika sistem sudah digunakan.

2.3.3 Kualitas Perangkat Lunak

Pengujian dilakukan untuk mendapatkan perangkat lunak dengan kualitas yang baik. Kualitas adalah karakteristik atau atribut dari sesuatu. Atribut yang dimaksud dalam konteks tersebut adalah segala hal yang dapat menggambarkan atau menjelaskan tentang produk tersebut, baik secara kualitatif maupun kuantitatif. Misalnya, dalam konteks kualitas perangkat lunak, atribut dapat mencakup keandalan, kinerja, keamanan, kemudahan penggunaan, dan sebagainya. Atribut ini membantu dalam menentukan seberapa baik atau buruk

produk tersebut dalam memenuhi kebutuhan pengguna dan persyaratan yang ditetapkan. Kualitas perangkat lunak adalah suatu keadaan yang secara jelas menyatakan permintaan dari fungsi dan kinerja, yang secara eksplisit dituliskan ke dalam dokumen standar pembangunan dan secara implisit menyatakan karakteristik yang diharapkan oleh semua pengembang *software*. Kualitas perangkat lunak dipengaruhi oleh kepuasan pelanggan, dan penting untuk menjaga kualitasnya karena beberapa alasan sebagai berikut:

1. Menjamin kelangsungan hidup dalam industri perangkat lunak.
2. Mampu bersaing dengan produk perangkat lunak lainnya.
3. Penting dalam upaya pemasaran global.
4. Meminimalisir pengeluaran dengan tujuan mengurangi pemborosan dalam bentuk pembuangan perangkat lunak akibat kegagalan dalam proses pemasaran atau produksi.
5. Mempertahankan pelanggan dan meningkatkan profitabilitas (Rosa & Shalahuddin, 2014).

2.3.4 Web Browser

Web browser yang sering disebut juga sebagai *Internet Browser*, bertindak sebagai penghubung antara pengguna komputer dan dunia maya, memfasilitasi penjelajahan internet. *Internet browser* adalah sebuah aplikasi atau perangkat lunak yang digunakan untuk mengelola data yang diterima dari *World Wide Web* (dikenal juga sebagai *www*) ke komputer pengguna dan menampilkannya secara visual agar dapat dipahami dengan mudah oleh pengguna internet. Adapun *web browser* yang biasa digunakan diantaranya adalah *Google Chrome*, *Mozilla Firefox*, *Microsoft Edge*, *Safari* (dikembangkan oleh Apple), *Opera*, *Brave Browser*, *Vivaldi* dan *UC Browser* (Uminingsih dkk., 2022).

2.3.5 Website

Sebuah *website* merupakan sebuah *platform* yang terdiri dari berbagai halaman yang terhubung satu sama lain melalui *hyperlink*. *Website* berfungsi sebagai media untuk menyajikan informasi dalam berbagai bentuk, seperti teks, gambar, video, suara, dan animasi, atau gabungan dari semua elemen tersebut. Karakteristik

utama dari sebuah *website* adalah kemampuannya untuk menyajikan halaman-halaman yang terhubung satu sama lain, serta memiliki alamat domain (URL) atau *World Wide Web* (www), dan tempat penyimpanan data. Dalam proses pengembangan sebuah *website* hingga tahap publikasi di internet, diperlukan beberapa aplikasi seperti database (misalnya MySQL, Oracle, dll), *Web Server* Apache, PHP Editor (seperti Macromedia, Notepad++, dll), dan Browser (Elgamar, 2020).

2.3.6 Blackbox Testing

Metode *Blackbox Testing* adalah pengujian dilakukan berdasarkan spesifikasi eksternal perangkat lunak dan tidak memeriksa struktur atau logika internal sebuah teknik pengujian perangkat lunak yang bertujuan untuk mengidentifikasi kesalahan atau kegagalan dalam fungsi-fungsi dan fitur-fitur sistem aplikasi tanpa memperhatikan detail internal dari struktur atau implementasi kode yang ada di dalamnya (Pressman, 2014). Dalam pengujian ini, fokus utama adalah pada interaksi luaran dengan sistem aplikasi, yang mencakup pengujian berbagai fungsi, menu, dan fitur-fitur aplikasi untuk memastikan bahwa semuanya berjalan sesuai yang diharapkan. Dengan kata lain, *Black Box Testing* bertujuan untuk memastikan bahwa sistem aplikasi berfungsi dengan benar dari perspektif pengguna atau pemakai layanan tanpa memperhatikan bagaimana kode atau logika internal sistem diimplementasikan. Ini adalah metode pengujian yang efektif dalam memvalidasi fungsionalitas aplikasi dari sudut pandang pengguna akhir dan memastikan bahwa semua fitur yang diharapkan berjalan dengan baik. Teknik pengujian pada *Black Box Testing* ada banyak macamnya yaitu:

- a. Teknik *Equivalence Partitioning* yaitu metode pengujian perangkat lunak yang digunakan untuk membagi input data menjadi beberapa partisi atau kelompok yang setara secara fungsional. Ide di balik teknik ini adalah untuk dapat memastikan bahwa perangkat lunak berfungsi dengan baik di satu bagian atau partisi dari input data, dengan demikian kemungkinan besar akan berfungsi dengan baik juga di bagian lain yang setara secara fungsional. Menggunakan teknik *Equivalence Partitioning* dalam pengujian dapat meminimalkan jumlah tes yang diperlukan untuk mencakup seluruh

ruang input yang mungkin, sambil tetap memastikan bahwa semua kasus uji yang relevan telah diuji. Teknik ini membantu dalam meningkatkan efisiensi pengujian perangkat lunak dengan memfokuskan perhatian pada kelas kesetaraan input yang berbeda.

- b. Teknik *Boundary Value Analysis* yaitu dengan cara mencari adakah *error* dari luar atau sisi dalam *software*, minimum maupun maksimum nilai dari *error* yang ditemukan atau dengan kata lain fokus pada nilai-nilai batas atau tepi dari rentang input. Tujuannya adalah untuk menemukan *error* yang mungkin terjadi pada batas minimal dan maksimal dari input yang diterima oleh perangkat lunak.
- c. Teknik *Fuzzing* yaitu merupakan teknik pengujian perangkat lunak yang bertujuan untuk menemukan bug atau kelemahan dalam sebuah program dengan cara menyuntikkan data yang tidak valid atau tidak terduga ke dalamnya. Ide di balik *Fuzzing* adalah dengan memberikan input yang tidak biasa atau tidak sesuai ke program, kita dapat menguji bagaimana program tersebut bereaksi dan apakah program tersebut rentan terhadap kesalahan atau kelemahan tertentu. Teknik *Fuzzing* sering digunakan dalam pengujian keamanan perangkat lunak untuk menemukan kerentanan atau celah keamanan yang dapat dieksploitasi oleh penyerang. Dengan memberikan input yang tidak valid, penguji dapat menemukan dan memperbaiki kerentanan keamanan sebelum perangkat lunak dirilis ke publik.
- d. Teknik *Cause-Effect Graph* adalah suatu teknik merancang *test case* berdasarkan hubungan sebab-akibat antara kondisi input (sebab) dan hasil yang diharapkan (akibat) melibatkan pemahaman yang mendalam tentang bagaimana input tertentu mempengaruhi perilaku atau output dari sistem yang diuji. Proses ini seringkali melibatkan analisis logika dari hubungan antara kondisi input dan efek yang diinginkan, serta menyusun *test case* yang mencakup berbagai kombinasi input untuk menguji berbagai skenario yang mungkin terjadi. Tujuan dari teknik ini adalah untuk memastikan bahwa sistem atau aplikasi berperilaku sesuai dengan spesifikasi yang telah ditetapkan.

- e. Teknik *All Pair Testing* yaitu cara untuk merencanakan *test case* sehingga semua kemungkinan kombinasi dari dua variabel atau parameter diuji minimal satu kali. Ini dilakukan dengan memilih pasangan dari nilai-nilai yang berbeda dari masing-masing parameter dan membuat *test case* yang mencakup semua pasangan ini. Tujuan testing ini adalah memiliki pasangan *test case* yang mencakup semua pasangan tersebut (Beser, 2008).
- f. Teknik *Error Guessing* yaitu teknik pengujian di mana penguji membuat daftar potensi kesalahan yang mungkin terjadi dalam aplikasi. Kemudian, pengujian dilakukan dengan mengikuti alur sesuai dengan daftar yang telah dibuat. Tujuan dari teknik *Error Guessing* dalam pengujian perangkat lunak adalah untuk menemukan kesalahan atau masalah yang mungkin terjadi dalam aplikasi, dengan menggunakan intuisi, pengalaman, dan pengetahuan tentang sistem. Dengan melakukan asumsi dan menebak potensi kesalahan, penguji dapat merancang *test case* yang relevan untuk menguji aplikasi secara lebih menyeluruh. Dengan demikian, tujuan utama dari *Error Guessing/Ad Hoc Testing* adalah untuk membantu memastikan kualitas dan kehandalan aplikasi dengan mengidentifikasi dan memperbaiki kesalahan sebelum aplikasi tersebut dirilis ke pengguna akhir (Jimmy dkk., 2021).

Dalam penelitian ini teknik *blackbox testing* yang akan digunakan sebagai acuan pembuatan *test case* untuk menguji *website* Disnaker diantaranya adalah teknik *Error Guessing*, *Boundary Value Analysis*, *Equivalence Partitioning* dan *Fuzzing*.

2.3.7 Software Testing Life Cycle (STLC)

Software Testing Life Cycle (STLC) atau siklus pengujian perangkat lunak adalah serangkaian langkah yang diatur secara terencana dan sistematis untuk menguji perangkat lunak. Dalam proses Siklus Hidup Pengujian Perangkat Lunak, sejumlah kegiatan dilakukan dengan maksud meningkatkan mutu produk. STLC merujuk pada serangkaian langkah khusus yang harus dijalani dalam pengujian untuk memastikan produk memiliki kualitas yang baik. Setiap tahap-tahap dari

STLC memiliki *Entry Criteria* (ketentuan yang harus dimiliki sebelum pengujian dimulai), *Exit Criteria* (ketentuan yang harus diselesaikan sebelum menyelesaikan pengujian), *Activities* dan *Deliverable* (hasil setelah melakukan pengujian) yang terkait (Dhaifullah dkk., 2022). Tahapan STLC dimulai dari *Requirement Analysis*, *Test Planning*, *Test Case Development*, *Environment Setup*, *Test Execution*, hingga *Test Cycle Closure*.

1. *Requirement Analysis*

Tahapan pertama yang dilakukan dalam *Software Testing Life Cycle* (STLC) yaitu menganalisis kebutuhan. Fase Analisis merupakan fase dasar dari proses pengujian perangkat lunak. Fase ini mencakup analisis kebutuhan fungsional dan non-fungsional misalnya persyaratan bisnis, dokumen spesifikasi fungsional dan dokumen spesifikasi teknis, dan lain-lain. Pengumpulan persyaratan dan harus dilakukan untuk penjelasan dengan pelanggan guna mengidentifikasi hasil aktual dan yang diharapkan dari pengujian. Tahap ini dilakukan dengan melakukan beberapa kegiatan guna menghasilkan dokumen kebutuhan pengguna. Proses memahami persyaratan secara rinci dilakukan dengan cara berinteraksi dengan CEO PT Indotechpren, *Project Manager*, serta para *developer* yang mengerjakan *project website* Disnaker.

2. *Test Planning*

Test planning merupakan tahapan persiapan ujian mencakup beberapa aspek kunci seperti penyusunan rencana ujian, pembuatan kasus ujian, pengadaan data ujian, dan persiapan lingkungan ujian. Dalam tahap perencanaan pengujian, fokusnya adalah merancang metode pengujian yang paling sesuai dan merencanakan upaya pengujian mulai dari penentuan waktu pengujian hingga sistem yang akan diuji. Rincian setiap pengujian dituangkan pada siklus STLC. Pada tahap ini dikumpulkan semua syarat yang diperlukan seperti analisis dokumen spesifikasi perangkat lunak untuk mengetahui kebutuhan sistem dan memperkirakan biaya yang mungkin dibutuhkan, serta menyiapkan strategi pengujian. Hasil dari proses ini adalah rencana pengujian (*test plan*) yang akan

menjadi dasar untuk membuat kasus uji (*test case*), serta perkiraan waktu yang diperlukan untuk melakukan pengujian. Setelah itu, tester dapat memulai proses pengembangan butir uji. Poin-poin dalam *test planning* sebagai berikut:

- a. Menetapkan ruang lingkup proyek pengujian .
- b. Mengidentifikasi daftar jenis pengujian.
- c. Menetapkan metode yang akan digunakan diantaranya menggunakan *automation testing tools* atau pengujian secara otomatis.
- d. Menentukan lingkungan pengujian.
- e. Menyusun langkah-langkah kegiatan atau prosedur pengujian.

3. *Test Case Development*

Test Case Development merupakan tahap yang melibatkan proses pembuatan, verifikasi, dan pengerjaan kasus uji. Dimulai dengan identifikasi data uji yang kemudian diperiksa dan dikerjakan kembali sesuai dengan persyaratan yang telah ditetapkan. Berikut adalah rangkaian aktivitas yang dilakukan dalam tahap ini:

- a. Membuat butir uji (*test case*) .
- b. Melakukan peninjauan terhadap kasus yang telah dibuat.
- c. Menyusun data pengujian.

Setelah tahap ini selesai, hasil yang diperoleh adalah *test case* dan data uji yang dapat digunakan dalam pengujian.

4. *Environment Setup*

Pada dasarnya, tahap *environment setup* bertujuan untuk memeriksa apakah lingkungan uji yang akan digunakan, baik perangkat keras maupun perangkat lunak telah beroperasi sesuai rencana yang telah ditetapkan. Fase ini dapat dilakukan secara bersamaan dengan fase pengembangan kasus uji (*test case development*). Tahap ini berdiri sendiri dan tidak terkait dengan fase lainnya. Berikut adalah aktivitas yang dilakukan dalam fase ini :

- a. Memahami struktur sistem, pengaturan lingkungan, dan menyiapkan perangkat lunak dan keras yang diperlukan oleh lingkungan pengujian.
- b. Menyiapkan data uji dan konfigurasi lingkungan uji (*setup test environment*).

Hasil dari tahap ini adalah lingkungan pengujian yang siap digunakan beserta pengaturan data uji yang telah disiapkan.

5. *Test Execution*

Tahap ini adalah tahap eksekusi yang mengikuti rencana pengujian (*test plan*) dan kasus uji (*test case*) yang telah disiapkan sebelumnya. Hasil eksekusi dari kasus uji (*test case*) dapat dinyatakan sebagai berhasil (*pass*) atau gagal (*fail*). Kasus uji yang sukses dieksekusi sesuai dengan harapan akan diberi status *pass*, sementara yang tidak sesuai akan diberi status *fail*. Jika ada *bug* yang ditemukan, temuan tersebut akan dicatat dalam dokumen pengujian dan dilaporkan kepada tim pengembang untuk diperbaiki. Pengujian akan diulang setelah *bug* tersebut telah diperbaiki. Aktivitas yang dilakukan antara lain:

- a. Mengeksekusi kasus uji (*test case*) sesuai rencana yang telah disusun.
- b. Mendokumentasikan hasil pengujian untuk pembuatan laporan, termasuk laporan bug jika ditemukan.

Hasil yang diperoleh dari kegiatan ini mencakup hasil eksekusi dari kasus uji dan laporan tentang *bug* yang ditemukan.

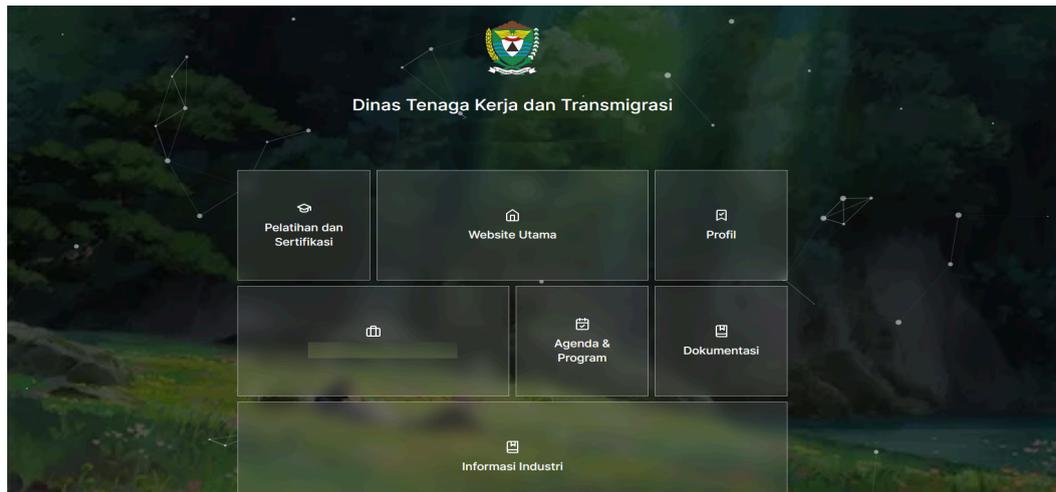
6. *Test Cycle Closure*

Test Cycle Closure merupakan tahap penutupan dari STLC. Tahap ini berisi tentang pelaporan penyelesaian tes, pengumpulan matriks penyelesaian tes dan hasil tes. Aktivitas yang dilakukan pada tahap ini sebagai berikut:

- a. Melakukan evaluasi dimensi karakteristik ISO/IEC 25010 yang telah diuji untuk mengetahui kualitas yang telah tercapai.
- b. Melakukan dokumentasi pengujian yang berisi hasil dari pengujian yang telah dilakukan.

2.3.8 Disnaker

Ini mungkin mencakup informasi tentang kebijakan, program-program pemerintah, layanan-layanan yang tersedia, dan sebagainya. *Website* ini memiliki menu diantaranya *home*, profil, dokumentasi, agenda dan program, bantuan, informasi serta validasi.



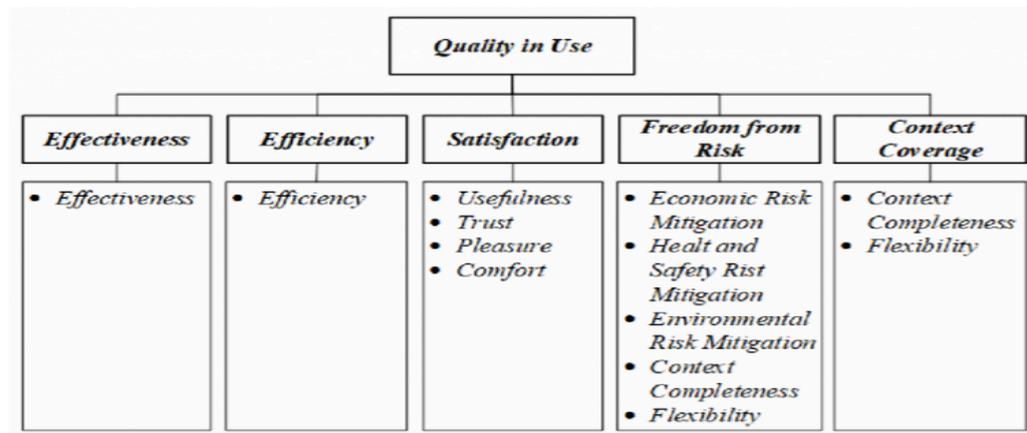
Gambar 2. Tampilan Utama *Website* Disnaker (disnakertrans.kab.go.id/menu).

Gambar 2 mengilustrasikan tampilan halaman utama dari *website* Disnaker. Pada halaman tersebut tampak beberapa menu utama yaitu Pelatihan dan Sertifikasi, *Website* Utama, Profil, Agenda dan Program, Dokumentasi serta Informasi Industri. Jika menu tersebut di klik akan mengarahkan pengguna ke masing-masing halaman sesuai dengan menu tersebut.

2.3.9 ISO/IEC 25010

ISO/IEC 25010 adalah standar internasional yang dapat digunakan untuk menganalisis, menguji, dan mengukur kualitas sistem perangkat lunak. Standar ini merupakan pembaruan dari ISO/IEC 9126 dengan penambahan struktur standar pada model kualitas. ISO/IEC 25010 telah menjadi standar tolak ukur untuk menganalisis sejauh mana kualitas perangkat lunak yang dipergunakan oleh perusahaan, organisasi, atau institusi tertentu. Menurut *International Organization for Standardization (ISO)* and *International Electrotechnical Commission (IEC)*, ISO/IEC 25010 memungkinkan evaluasi kualitas sistem perangkat lunak secara spesifik berdasarkan dua dimensi, yaitu *Quality in Use* dan *Software Product*

Quality. Dimensi *Quality in Use* menggambarkan efek interaksi sistem terhadap pengguna atau pemangku kepentingan.

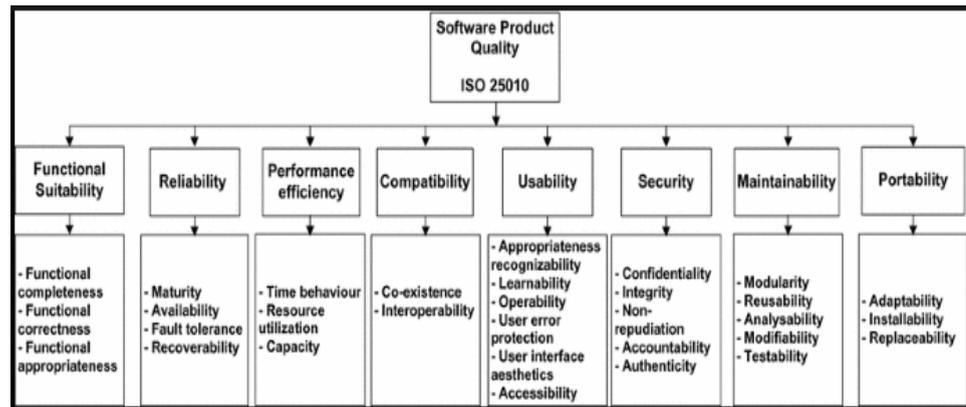


Gambar 3. Dimensi *Quality in Use* ISO/IEC 25010.

Gambar 3 merupakan ilustrasi dimensi *Quality in Use* ISO/IEC 25010. Terdapat lima karakteristik pada dimensi *Quality in Use* ISO/IEC 25010 diantaranya :

- 1) Keefektifan (*Effectiveness*), karakteristik yang digunakan untuk menilai kemampuan sistem untuk mencapai tujuan dengan akurasi dan integritas.
- 2) Efisiensi (*Efficiency*), karakteristik yang digunakan untuk mengukur kemampuan sistem dalam menggunakan sumber daya dengan tepat saat mencapai tujuan dengan akurasi dan integritas.
- 3) *Satisfaction*, karakteristik yang digunakan untuk mengukur kemampuan sistem untuk memenuhi kebutuhan pengguna. Karakteristik ini terdiri dari sub karakteristik:
 - a) Kebermanfaatan (*Usefulness*): Kemampuan sistem untuk memenuhi kebutuhan praktis pengguna.
 - b) Kepercayaan (*Trust*): Kemampuan sistem untuk membuat pengguna atau pemangku kepentingan percaya pada perilaku sistem.
 - c) Kesenangan (*Pleasure*): Kemampuan sistem untuk memberikan pengguna pengalaman yang menyenangkan saat digunakan untuk memenuhi kebutuhan.
 - d) Kenyamanan (*Comfort*): Sejauh mana sistem memberikan kenyamanan bagi pengguna saat digunakan.

- 4) *Freedom from Risk*, karakteristik yang digunakan untuk mengukur kemampuan sistem untuk mengurangi risiko. karakteristik ini terdiri dari sub karakteristik:
 - a) Mitigasi risiko ekonomi (*Economic risk mitigation*): Kemampuan sistem untuk mengurangi risiko terkait keuangan, operasional, dan sumber daya lainnya yang terkait dengan penggunaan sistem.
 - b) Mitigasi risiko kesehatan dan keselamatan (*Health and safety risk mitigation*): Kemampuan sistem untuk mengurangi risiko terkait kesehatan dan keselamatan pengguna saat menggunakan sistem.
 - c) Mitigasi risiko lingkungan (*Environmental risk mitigation*): Kemampuan sistem untuk mengurangi risiko terkait lingkungan saat penggunaan sistem.
- 5) *Context Completeness*, karakteristik yang menilai kemampuan sistem dalam penggunaan yang efektif, efisien, dan bebas risiko. Karakteristik ini mencakup sub-karakteristik berikut:
 - a) Kelengkapan Konteks (*Context completeness*): Kemampuan sistem untuk digunakan secara efektif, efisien, bebas risiko, dan memberikan kepuasan dalam penggunaan sesuai dengan konteks penggunaannya.
 - b) Fleksibilitas (*Flexibility*): Kemampuan sistem untuk digunakan secara efektif, efisien, bebas risiko, dan memberikan kepuasan di luar batasan persyaratan yang telah ditetapkan, sesuai dengan kebutuhan dan situasi penggunaannya.



Gambar 4. Dimensi *Software Product Quality* ISO/IEC 25010.

Gambar diatas merupakan ilustrasi dimensi *Software Product Quality* ISO/IEC 25010. Terdapat delapan karakteristik pada dimensi *Software Product Quality* ISO/IEC 25010 antara lain:

- 1) *Functional Suitability*, karakteristik yang digunakan untuk mengukur seberapa baik sistem dapat memenuhi kebutuhan pengguna saat digunakan. Di mana karakteristik ini terdiri dari sub karakteristik:
 - a) *Functional completeness*, mengukur sejauh mana sistem mampu mencakup semua tugas dan tujuan yang telah ditetapkan oleh pengguna.
 - b) *Functional correctness*, kemampuan fungsi sistem dalam menilai kemampuan sistem untuk memberikan hasil yang tepat sesuai dengan tingkat akurasi yang diharapkan.
 - c) *Functional appropriateness*, kemampuan fungsi sistem dalam mendukung pencapaian tugas dan tujuan yang telah ditetapkan dengan tepat.
- 2) *Performance Efficiency*, karakteristik yang digunakan untuk karakteristik yang digunakan untuk melakukan pengukuran terhadap kinerja sistem berdasarkan penggunaan sumber daya yang digunakan. Karakteristik ini terdiri dari sub karakteristik:
 - a) *Time behaviour*, kemampuan sistem dalam memproses dan merespon dengan cepat ketika menjalankan fungsinya.

- b) *Resource utilization*, kemampuan sistem dalam efisien menggunakan jumlah dan jenis sumber daya yang dibutuhkan saat menjalankan fungsinya.
 - c) *Capacity*, kemampuan sistem untuk menangani beban kerja maksimum dan mempertahankan kinerja yang baik ketika menjalankan fungsinya.
- 3) *Compatibility*, karakteristik yang digunakan untuk mengukur kemampuan dua atau lebih sistem dalam bertukar informasi dan melakukan fungsi dalam lingkungan hardware maupun software yang sama. Karakteristik ini terdiri dari sub karakteristik:
- a) *Co-existence*, kemampuan sistem untuk beroperasi secara efisien dan berbagi lingkungan serta sumber daya yang sama dengan sistem lain tanpa mengganggu kinerja atau merugikan sistem lain.
 - b) *Interoperability*, kemampuan dua atau lebih sistem untuk berbagi dan menggunakan informasi yang telah dibagikan dengan efektif.
- 4) *Usability*, karakteristik yang mengukur kemampuan sistem dalam operasi dan kontrol oleh pengguna dengan efisiensi, efektivitas, dan tingkat kepuasan yang tinggi. Karakteristik ini terdiri dari sub-karakteristik berikut:
- a) *Appropriateness recognize ability*, kemampuan sistem untuk dikenali oleh pengguna sesuai dengan kebutuhan dan tujuannya.
 - b) *Learnability*, kemampuan sistem untuk dipelajari dengan mudah dan efektif oleh pengguna, serta memberikan kepuasan terhadap tujuan penggunaan tertentu.
 - c) *Operability*, kemampuan sistem untuk dioperasikan dan dikendalikan dengan mudah oleh pengguna.
 - d) *User error protection*, kemampuan sistem untuk melindungi pengguna dari kesalahan yang mungkin dilakukan.
 - e) *User interface aesthetics*, kemampuan antarmuka pengguna untuk memberikan kesenangan dan kepuasan bagi pengguna.

- f) *Accessibility*, kemampuan sistem untuk digunakan oleh pengguna dengan berbagai karakteristik, serta untuk mencapai tujuan penggunaan yang spesifik.
- 5) *Reliability*, karakteristik yang digunakan untuk mengevaluasi kemampuan sistem dalam menjaga kinerjanya saat digunakan dalam kondisi dan periode waktu tertentu. Karakteristik ini mencakup sub-karakteristik berikut:
- a) *Kematangan (Maturity)*: Kemampuan sistem untuk memenuhi kebutuhan akan keandalan dalam operasional normalnya.
 - b) *Ketersediaan (Availability)*: Kemampuan sistem untuk tetap dapat dioperasikan dan diakses saat diperlukan.
 - c) *Toleransi terhadap Kesalahan (Fault tolerance)*: Kemampuan sistem untuk tetap beroperasi dengan baik meskipun terjadi kesalahan pada perangkat keras atau perangkat lunaknya.
 - d) *Recoverability*, karakteristik yang menilai kemampuan sistem untuk mengembalikan data yang terpengaruh secara langsung dan mengembalikan sistem ke kondisi yang diinginkan.
- 6) *Security*, karakteristik yang mengukur kemampuan sistem dalam melindungi informasi dan data dari akses, penggunaan, modifikasi, perusakan, serta tindakan jahat yang disengaja atau tidak disengaja. Karakteristik ini terdiri dari beberapa sub-karakteristik, yaitu:
- a) *Confidentiality*, kemampuan sistem untuk memastikan bahwa hanya pengguna yang berwenang yang dapat mengakses data.
 - b) *Integrity*, kemampuan sistem untuk mencegah akses yang tidak sah atau modifikasi terhadap program atau data..
 - c) *Non-repudiation*, kemampuan sistem untuk membuktikan peristiwa atau tindakan yang telah terjadi sehingga tidak dapat disangkal.
 - d) *Accountability*, kemampuan sistem untuk melacak tindakan entitas secara unik..

- e) *Authenticity*, kemampuan sistem untuk mengidentifikasi subjek atau sumber daya yang telah diverifikasi sebagai keaslian atau klaim yang sah.
- 7) *Maintainability*, karakteristik yang digunakan untuk mengukur kemampuan sistem untuk dilakukan modifikasi mencakup pengoreksian, perbaikan, serta penyesuaian perubahan lingkungan, persyaratan, dan spesifikasi fungsional sistem. Karakteristik ini terdiri dari sub karakteristik:
- a) *Modularity* yaitu kemampuan sistem untuk berinteraksi dengan komponen-komponennya secara terpisah, sehingga perubahan pada satu komponen tidak berdampak besar pada komponen lainnya.
 - b) *Reusability*, kemampuan sistem untuk menggunakan kembali asetnya pada sistem lain atau dalam membangun aset baru.
 - c) *Analysability*, kemampuan sistem untuk mendiagnosis atau mengidentifikasi tingkat efektivitas dan efisiensi terhadap dampak perubahan sistem pada satu atau lebih bagian, serta kekurangan atau pemicu kegagalan terhadap bagian-bagian yang akan diperbaiki.
 - d) *Modifiability*, kemampuan sistem untuk mengalami modifikasi dengan efektif tanpa mengakibatkan kekurangan atau penurunan kualitas sistem yang ada.
 - e) *Testability*, kemampuan sistem untuk diuji secara efektif dan efisien berdasarkan kriteria pengujian yang telah ditetapkan. Pengujian ini bertujuan untuk menentukan apakah kriteria-kriteria tersebut telah terpenuhi.
- 8) *Portability*, karakteristik yang mengukur kemampuan sistem untuk dipindahkan atau ditransfer ke lingkungan sistem lain, termasuk hardware, software, operasional, atau penggunaan lainnya. Karakteristik ini terdiri dari beberapa sub-karakteristik, yaitu:

- a) *Adaptability*, kemampuan sistem untuk disesuaikan secara efektif dan efisien dengan hardware, software, atau lingkungan operasional atau penggunaan lain yang berbeda atau berkembang.
- b) *Installability*, kemampuan sistem untuk diinstal atau dihapus dengan tingkat efektivitas dan efisiensi tertentu dalam lingkungan yang spesifik.
- c) *Replaceability*, kemampuan sistem untuk menggantikan sistem perangkat lunak lain yang memiliki tujuan dan lingkungan yang serupa atau sama (British Standards Institution, 2011).

Dalam penelitian ini dimensi ISO/IEC 25010 yang akan digunakan adalah model *Software Product Quality* dengan karakteristik *Functional Suitability*, *Performance Efficiency*, *Usability* dan *Portability*.

2.3.10 Karakteristik *Usability* ISO/IEC 25010

Karakteristik *Usability* merupakan karakteristik dalam pengujian perangkat lunak yang digunakan untuk menilai seberapa baik suatu produk atau sistem dapat dimanfaatkan oleh pengguna secara efektif, efisien, dan memuaskan. Definisi tersebut memfokuskan pada aspek perspektif pengguna dalam pengujian *usability*. Pengujian akan dilakukan pada karakteristik *Usability*, kegiatan tersebut biasa disebut dengan *Usability Testing*. Dalam rangkaian pengujian perangkat lunak, *Usability* merupakan tahap yang melibatkan pengguna akhir sebagai partisipan utama. Partisipan diminta untuk menjalankan serangkaian tugas yang diberikan terkait *website* yang diuji, sementara alat pengujian perangkat lunak Maze dan Useberry akan digunakan untuk mencatat dan mengamati interaksi partisipan dengan *website* yang sedang diuji. Pengujian dilakukan untuk mengukur sejauh mana suatu produk atau sistem dapat digunakan dengan mudah dan efektif berdasarkan penilaian pengguna (Wicaksono, 2023).

Usability Testing memiliki peranan penting dalam proses evaluasi perangkat lunak, berfungsi meningkatkan kualitas pengalaman pengguna dan meminimalisir kemungkinan kegagalan produk. Identifikasi masalah *usability* memungkinkan pengembang untuk menyempurnakan desain dan fungsionalitas perangkat lunak, sehingga lebih efektif dan mudah dioperasikan oleh pengguna. Selain itu, tim

pengembang dapat memperoleh pemahaman yang lebih baik tentang preferensi dan kebutuhan pengguna, sehingga dapat menghasilkan produk yang lebih sesuai dengan tuntutan pasar. Dalam lingkup pengujian perangkat lunak, pendekatan ini juga berpotensi menghemat biaya dan meningkatkan efisiensi proses pengembangan secara keseluruhan. Di era transformasi digital yang kian pesat, pengujian kebergunaan memegang peranan yang semakin penting dalam proses pengujian perangkat lunak. Ekspektasi pengguna terhadap pengalaman yang lebih mudah dan efisien terus meningkat, menyebabkan perangkat lunak yang tidak memenuhi standar kegunaan berisiko kehilangan pengguna dan kredibilitas. Dengan menjalankan *usability testing* secara berkala, tim pengembang dapat memastikan bahwa produk yang dihasilkan tetap adaptif terhadap perubahan kebutuhan dan harapan pengguna (Wicaksono, 2023).

Tahapan untuk melakukan *usability testing* mengikuti tahapan umum pada pengujian perangkat lunak atau yang sering disebut dengan siklus hidup pengujian perangkat lunak (*Software Testing Life Cycle*) sebagai berikut :

1. Requirement Analysis

Requirement analysis merupakan tahapan dalam pengujian perangkat lunak yang berfokus pada pemahaman mendalam tentang kebutuhan sistem atau aplikasi yang akan diuji. Adapun kegiatan yang akan dilakukan pada tahap ini adalah sebagai berikut :

- a. Mengidentifikasi tujuan melakukan *usability testing*.
- b. Mengidentifikasi profil atau karakteristik pengguna.
- c. Mengidentifikasi sub karakteristik yang akan diuji dalam *usability testing*.
- d. Mengidentifikasi metrik yang akan digunakan dalam pengujian.
- e. Mengidentifikasi tugas yang akan dilakukan oleh partisipan sebagai target *usability testing*.

2. Test Planning

Test Planning merupakan tahapan sistematis yang ada dalam pengujian perangkat lunak untuk merancang, mengorganisir, dan mendokumentasikan

strategi pengujian yang akan digunakan untuk mengevaluasi seberapa mudah dan efektif suatu produk atau antarmuka dapat digunakan oleh pengguna target. Ini merupakan tahap kritis dalam persiapan *usability testing* yang bertujuan untuk memastikan bahwa pengujian dilakukan secara terstruktur, efisien, dan menghasilkan data yang bermakna untuk meningkatkan pengalaman pengguna. Kegiatan yang dilakukan pada tahap *test planning* adalah sebagai berikut:

- a. Menetapkan ruang lingkup pengujian.
- b. Menetapkan partisipan sebagai target *usability testing*.
- c. Menetapkan metode yang akan digunakan dalam pengujian.
- d. Menentukan lingkungan pengujian.
- e. Menentukan tugas yang akan diberikan kepada partisipan dalam kegiatan pengujian.
- f. Menetapkan *tools* yang akan digunakan dalam melakukan pengujian.
- g. Menentukan indikator keberhasilan.

3. *Test Case Development*

Test Case Development merupakan tahapan merancang skenario pengujian yang realistis dan relevan dan tugas spesifik dirancang untuk mengevaluasi kegunaan (*usability*) suatu produk atau sistem yang akan dilakukan oleh partisipan selama sesi *usability testing*. Tujuannya adalah untuk menciptakan situasi penggunaan yang mewakili interaksi pengguna nyata dengan perangkat lunak dalam hal ini adalah *website* Disnaker Kabupaten XYZ. Beberapa kegiatan yang dilakukan pada tahap ini diantaranya adalah sebagai berikut :

- a. Menentukan fitur yang akan diuji.
- b. Membangun *test case* yang dalam hal ini dituangkan dalam skenario atau tugas yang harus diselesaikan oleh partisipan menggunakan *tools*.

4. *Environment Setup*

Environment Setup merupakan tahapan dalam pengujian perangkat lunak dalam mempersiapkan lingkungan di mana pengujian akan dilakukan. Tujuannya adalah menciptakan kondisi yang optimal dan realistis untuk

mengevaluasi *usability* produk. Adapun kegiatan yang dilakukan pada tahap ini adalah sebagai berikut :

- a. Memastikan perangkat memiliki spesifikasi yang sesuai dengan pengguna target.
- b. Menyiapkan prototipe perangkat lunak yang akan diuji.
- c. Melakukan konfigurasi pada *tools* yang akan digunakan untuk pengujian.
- d. Melakukan tes awal untuk memastikan *tools* yang digunakan untuk melakukan pengujian telah berfungsi dengan baik.

5. Test Execution

Test Execution merupakan tahapan dalam pengujian perangkat lunak di mana skenario dan tugas yang telah dirancang sebelumnya dijalankan dengan partisipan yang sebenarnya. Ini adalah fase kritis di mana data aktual dikumpulkan. Kegiatan yang dilakukan pada tahap ini adalah sebagai berikut:

- a. Menyebarkan tautan *tools* Maze dan Useberry yang telah disiapkan kepada partisipan.
- b. Mengarahkan partisipan untuk melakukan pengujian.

6. Test Closure

Test Closure merupakan tahap akhir dalam proses *usability testing* di mana seluruh aktivitas pengujian disimpulkan, data dianalisis, dan hasil-hasil dikomunikasikan. Tahap ini penting untuk memastikan bahwa semua informasi yang dikumpulkan dimanfaatkan dengan baik dan menghasilkan insight yang berguna untuk perbaikan produk. Adapun kegiatan yang dilakukan pada tahap ini sebagai berikut :

- a. Menganalisis hasil yang diperoleh dari hasil pengujian yang telah dilakukan oleh partisipan.
- b. Menyimpulkan nilai atau hasil pengujian karakteristik *usability* yang telah dilakukan pada *tools* Maze dan Useberry.
- c. Melakukan evaluasi terhadap pengujian.

2.3.11 Karakteristik *Portability* ISO/IEC 25010

Portabilitas (*Portability*) adalah karakteristik yang menggambarkan kemampuan suatu sistem, produk, atau komponen untuk berpindah atau beradaptasi secara

efektif dan efisien antara berbagai lingkungan komputasi. Ini mencakup perpindahan antar perangkat keras, perangkat lunak, lingkungan operasional, atau konteks penggunaan yang berbeda. Mengutip Peter J. Brown dalam karya Ville Salonen (2012), perangkat lunak dianggap portabel jika dapat dioperasikan pada berbagai platform komputer tanpa perubahan signifikan. Idealnya, perangkat lunak portabel dapat ditransfer antar lingkungan tanpa memerlukan modifikasi pada kode sumbernya. Dalam konteks modern, khususnya untuk aplikasi web, portabilitas juga mencakup kemampuan perangkat lunak untuk berfungsi secara konsisten di berbagai jenis browser, baik pada perangkat desktop maupun mobile. Konsep portabilitas ini menegaskan fleksibilitas dan adaptabilitas sebagai aspek kunci dalam pengembangan perangkat lunak, memungkinkan penggunaan yang lebih luas dan efisien di berbagai lingkungan teknologi (Salonen, 2012). Penting untuk melakukan pengujian pada karakteristik portabilitas. Kegiatan pengujian portabilitas biasa disebut dengan *Portability Testing*. Karakteristik ini memiliki sub karakteristik sebagai berikut:

1. *Adaptability*, merujuk pada fleksibilitas sistem dalam menyesuaikan diri dengan berbagai kondisi. Ini mencakup kemampuan untuk berfungsi secara optimal di berbagai konfigurasi perangkat keras, platform perangkat lunak, atau dalam konteks operasional yang beragam. Adaptabilitas juga meliputi kemampuan sistem untuk mengikuti perkembangan teknologi tanpa kehilangan fungsionalitas utamanya.
2. *Installability*, mengacu pada tingkat kemudahan dan efisiensi sebuah sistem dalam proses pemasangan dan penghapusannya pada suatu lingkungan komputasi tertentu. Karakteristik ini menilai seberapa lancar dan efektif sistem dapat diintegrasikan ke dalam atau dihapus dari sebuah lingkungan operasional spesifik.
3. *Replaceability*, menilai sejauh mana sistem dapat menggantikan perangkat lunak lain yang memiliki fungsi serupa dalam lingkungan yang sama atau mirip. Ini melibatkan kompatibilitas dengan sistem yang ada, kemampuan untuk mengadopsi fungsi-fungsi kritis dari sistem sebelumnya, dan meminimalkan gangguan selama proses transisi (British Standards Institution, 2011).

Tahapan untuk melakukan *portability testing* mengikuti tahapan umum pada pengujian perangkat lunak atau yang sering disebut dengan siklus hidup pengujian perangkat lunak (*Software Testing Life Cycle*) sebagai berikut :

1. Requirement Analysis

Requirement analysis merupakan tahapan dalam pengujian perangkat lunak yang berfokus pada mengidentifikasi, memahami, dan mendokumentasikan semua kebutuhan terkait portabilitas sebuah aplikasi atau sistem. Tujuan utamanya adalah untuk memastikan bahwa perangkat lunak dapat berfungsi dengan baik di berbagai lingkungan dan platform yang berbeda. Adapun kegiatan yang akan dilakukan pada tahap ini adalah sebagai berikut :

- a. Mengidentifikasi tujuan melakukan *portability testing*.
- b. Mengidentifikasi sub karakteristik yang akan diuji dalam *portability testing*.
- c. Mengidentifikasi metrik yang akan digunakan dalam pengujian.
- d. Mengidentifikasi *tools* yang akan digunakan untuk pengujian.

2. Test Planning

Test Planning merupakan tahapan merencanakan dan merancang strategi untuk menguji portabilitas perangkat lunak. Tujuannya adalah memastikan pengujian dilakukan secara terstruktur, efisien, dan mencakup semua aspek portabilitas yang telah diidentifikasi dalam tahap *Requirement Analysis*. Kegiatan yang dilakukan pada tahap *test planning* adalah sebagai berikut:

- a. Menetapkan ruang lingkup pengujian.
- b. Menetapkan metode yang akan digunakan dalam pengujian.
- c. Menentukan indikator keberhasilan.
- d. Menetapkan *tools* yang akan digunakan dalam melakukan pengujian.

3. Test Case Development

Test Case Development merupakan tahapan pengujian merancang dan menyusun skenario pengujian spesifik untuk mengevaluasi portabilitas *website* Disnaker Kabupaten XYZ. Tujuannya adalah menciptakan serangkaian *test case* yang komprehensif untuk memverifikasi bahwa aplikasi berfungsi dengan baik di berbagai lingkungan dan platform. Beberapa kegiatan yang dilakukan pada tahap ini diantaranya adalah sebagai berikut :

- a. Menentukan fitur yang akan diuji.
- b. Membangun *test case* untuk setiap kombinasi sistem operasi, perangkat, dan browser yang telah diidentifikasi.

4. *Environment Setup*

Environment Setup merupakan tahapan dalam pengujian perangkat lunak dalam mempersiapkan dan mengkonfigurasi berbagai lingkungan yang diperlukan untuk melaksanakan pengujian portabilitas secara efektif. Tujuannya adalah untuk menciptakan kondisi yang merepresentasikan berbagai platform dan konfigurasi di mana aplikasi diharapkan dapat berfungsi. Adapun kegiatan yang dilakukan pada tahap ini adalah sebagai berikut :

- a. Melakukan konfigurasi pada *tools* yang akan digunakan untuk pengujian.
- b. Melakukan tes awal untuk memastikan *tools* yang digunakan untuk melakukan pengujian telah berfungsi dengan baik.

5. *Test Execution*

Test Execution merupakan tahapan dalam pengujian perangkat lunak di mana skenario dan tugas yang telah dirancang sebelumnya dijalankan pada berbagai lingkungan dan platform yang telah disiapkan. Tujuannya adalah untuk memverifikasi portabilitas aplikasi dan mengidentifikasi masalah yang mungkin muncul di berbagai kondisi penggunaan. Kegiatan yang dilakukan pada tahap ini adalah menjalankan pengujian di setiap kombinasi platform, perangkat, dan browser yang telah diidentifikasi.

6. *Test Closure*

Test Closure merupakan tahap akhir dalam proses *portability testing* di mana seluruh aktivitas menyelesaikan aktivitas pengujian, menganalisis hasil secara keseluruhan, dan menyimpulkan status portabilitas *website* Disnaker Kabupaten XYZ. Adapun kegiatan yang dilakukan pada tahap ini sebagai berikut :

- a. Menganalisis hasil pengujian .
- b. Menyimpulkan dan mengevaluasi hasil pengujian karakteristik *portability* yang telah dilakukan pada Browserstack.

2.3.12 Karakteristik *Fungsional Suitability* ISO/IEC 25010

Karakteristik *Fungsional Suitability* adalah salah satu karakteristik kualitas perangkat lunak yang didefinisikan dalam standar ISO/IEC 25010. Ini merupakan bagian dari model kualitas produk perangkat lunak yang menilai sejauh mana suatu produk atau sistem menyediakan fungsi-fungsi yang memenuhi kebutuhan yang dinyatakan dan tersirat ketika digunakan dalam kondisi tertentu. Karakter ini akan diuji dengan pengujian fungsional (*Fungsional Testing*). Tahapan untuk melakukan *functional testing* mengikuti tahapan umum pada pengujian perangkat lunak atau yang sering disebut dengan siklus hidup pengujian perangkat lunak (*Software Testing Life Cycle*) sebagai berikut :

1. *Requirement Analysis*

Requirement analysis merupakan tahapan dalam pengujian perangkat lunak yang berfokus pada pemahaman mendalam tentang kebutuhan sistem atau aplikasi yang akan diuji dari segi fungsionalitas. Adapun kegiatan yang akan dilakukan pada tahap ini adalah sebagai berikut :

- a. Mengidentifikasi tujuan melakukan *fungsional testing*.
- b. Mengidentifikasi kebutuhan fungsional dan non fungsional.
- c. Mengidentifikasi profil atau karakteristik pengguna akhir.
- d. Mengidentifikasi sub karakteristik yang akan diuji dalam *fungsional testing*.
- e. Mengidentifikasi metrik yang akan digunakan dalam pengujian.

2. *Test Planning*

Test Planning merupakan tahapan sistematis yang ada dalam pengujian perangkat lunak untuk merancang, mengorganisir, dan mendokumentasikan strategi pengujian yang mencakup semua aspek penting dari fungsionalitas. Ini merupakan tahap kritis dalam persiapan *fungsional testing* yang bertujuan untuk memastikan bahwa pengujian dilakukan secara terstruktur, efisien, dan maksimal dalam menemukan kesalahan. Kegiatan yang dilakukan pada tahap *test planning* adalah sebagai berikut:

- a. Menetapkan ruang lingkup pengujian.
- b. Menetapkan metode yang akan digunakan dalam pengujian
- c. Menetapkan *tools* yang akan digunakan dalam melakukan pengujian.
- d. Menentukan indikator keberhasilan.

3. Test Case Development

Test Case Development tahapan merancang skenario pengujian yang realistis dan relevan serta tugas spesifik untuk mengevaluasi fungsionalitas *website* Disnaker Kabupaten XYZ. Tujuannya adalah untuk menciptakan serangkaian *test case* yang komprehensif dan terstruktur untuk memverifikasi bahwa semua fitur dan fungsi *website* berjalan sesuai dengan spesifikasi yang telah ditentukan. Beberapa kegiatan yang dilakukan pada tahap ini diantaranya adalah sebagai berikut :

- a. Menentukan fitur yang akan diuji.
- b. Membangun *test case* menggunakan berbagai teknik yang ada pada metode *blackbox testing*.

4. Environment Setup

Environment Setup merupakan tahapan berfokus pada mempersiapkan lingkungan di mana pengujian fungsional akan dilakukan. Tujuannya adalah menciptakan kondisi yang optimal dan realistis untuk mengevaluasi fungsionalitas *website* Disnaker Kabupaten XYZ. Adapun kegiatan yang dilakukan pada tahap ini adalah sebagai berikut :

- a. Memastikan perangkat memiliki spesifikasi yang sesuai dengan *tools* yang akan digunakan untuk melakukan pengujian.
- b. Melakukan instalasi dan konfigurasi *tools* pengujian.
- c. Melakukan tes awal untuk memastikan *tools* yang digunakan untuk melakukan pengujian telah berfungsi dengan baik.

5. Test Execution

Test Execution merupakan tahapan dalam pengujian perangkat lunak di mana skenario dan *test case* yang telah dirancang sebelumnya dijalankan untuk menguji fungsionalitas *website* Disnaker Kabupaten XYZ. Ini adalah fase kritis di mana data aktual dikumpulkan untuk mengevaluasi kinerja dan keandalan sistem. Kegiatan yang dilakukan pada tahap ini adalah sebagai berikut :

- a. Mengeksekusi *test case* sesuai dengan urutan yang telah ditentukan.
- b. Mencatat hasil setiap langkah pengujian dengan teliti.

6. *Test Cycle Closure*

Test Closure merupakan tahap akhir dalam proses functional testing di mana seluruh aktivitas pengujian disimpulkan, data dianalisis, dan dievaluasi. Tahap ini penting untuk memastikan bahwa pengujian telah dilaksanakan dan mendapatkan hasil yang aktual. Adapun kegiatan yang dilakukan pada tahap ini sebagai berikut :

- a. Menganalisis hasil yang diperoleh dari hasil pengujian yang telah dilakukan.
- b. Menyimpulkan nilai atau hasil pengujian karakteristik *usability* yang telah dilakukan pada *tools* Selenium IDE dan Katalon Recorder.
- c. Melakukan evaluasi terhadap pengujian.

2.3.13 Karakteristik *Performance Efficiency* ISO/IEC 25010

Karakteristik *Performance Efficiency* adalah karakteristik yang mengukur kinerja suatu sistem atau perangkat lunak dalam konteks penggunaan sumber daya. Ini berfokus pada bagaimana sistem beroperasi secara efisien, memaksimalkan output sambil meminimalkan input sumber daya. *Performance Efficiency* mengacu pada tingkat kinerja perangkat lunak relatif terhadap jumlah sumber daya yang digunakan dalam kondisi yang telah ditentukan. Sumber daya ini dapat mencakup perangkat keras, perangkat lunak (seperti sistem operasi, database), atau material (seperti kertas untuk output cetak). Pentingnya *Performance Efficiency*:

- a. Pengalaman Pengguna: Kinerja yang baik meningkatkan kepuasan pengguna dan produktivitas.
- b. Skalabilitas: Memungkinkan sistem untuk menangani peningkatan beban atau pertumbuhan data.
- c. Efisiensi Biaya: Penggunaan sumber daya yang efisien dapat mengurangi biaya operasional.
- d. Keunggulan Kompetitif: Aplikasi yang berkinerja baik dapat memberikan keunggulan di pasar.

Kegiatan pengujian terhadap karakteristik *performance efficiency* biasa disebut dengan *performance testing*. Menurut Dini Yuniasri, *performance efficiency* diukur berdasarkan perilaku waktu atau *time behavior*, yaitu waktu respons sistem

saat memberikan hasil yang sesuai dengan permintaan pengguna (Yuniasri et al., 2020). Oleh karena itu, penilaian dilakukan berdasarkan *response time* yang diukur menggunakan JMeter dan Blazemeter. Norman & Nielsen (2010) menyatakan bahwa waktu respons yang dianggap baik seharusnya tidak melebihi 10 detik. Tahapan untuk melakukan *performance efficiency* mengikuti tahapan umum pada pengujian perangkat lunak atau yang sering disebut dengan siklus hidup pengujian perangkat lunak (*Software Testing Life Cycle*) sebagai berikut :

1. Requirement Analysis

Requirement analysis merupakan tahapan awal yang berfokus pada pemahaman mendalam tentang kebutuhan sistem atau aplikasi yang akan diuji dari segi kinerja. Adapun kegiatan yang akan dilakukan pada tahap ini adalah sebagai berikut :

- a. Mengidentifikasi tujuan melakukan *performance testing*.
- b. Mengidentifikasi sub karakteristik yang akan diuji dalam *performance testing*.
- c. Mengidentifikasi metrik yang akan digunakan dalam pengujian.
- d. Mengidentifikasi fitur dan role yang akan diuji.

2. Test Planning

Test Planning merupakan tahapan sistematis untuk merancang, mengorganisir, dan mendokumentasikan strategi pengujian yang mencakup semua aspek penting dari kinerja sistem. Ini merupakan tahap kritis dalam persiapan *performance testing* yang bertujuan untuk memastikan bahwa pengujian dilakukan secara terstruktur, efisien, dan efektif dalam mengukur dan mengevaluasi kinerja sistem. Kegiatan yang dilakukan pada tahap *test planning* adalah sebagai berikut:

- a. Menetapkan ruang lingkup pengujian.
- b. Menetapkan metode dan *tools* yang akan digunakan dalam pengujian.
- c. Menetapkan teknik pengujian yang akan digunakan.
- d. Menetapkan indikator keberhasilan.

3. *Test Case Development*

Test Case Development adalah tahapan merancang skenario pengujian yang realistis dan relevan serta tugas spesifik untuk mengevaluasi kinerja *website* Disnaker Kabupaten XYZ. Tujuannya adalah untuk menciptakan serangkaian test case yang komprehensif dan terstruktur untuk mengukur dan mengevaluasi kinerja sistem dalam berbagai kondisi beban. Beberapa kegiatan yang dilakukan pada tahap ini diantaranya adalah sebagai berikut :

- a. Menetapkan fitur yang akan diuji dalam *performance testing*.
- b. Membangun *test case* menggunakan berbagai teknik *performance testing*.

4. *Environment Setup*

Environment Setup merupakan tahapan yang berfokus pada mempersiapkan lingkungan di mana pengujian kinerja akan dilakukan. Tujuannya adalah menciptakan kondisi yang optimal dan realistis untuk mengevaluasi kinerja *website* Disnaker Kabupaten XYZ di bawah berbagai kondisi beban. Adapun kegiatan yang dilakukan pada tahap ini adalah sebagai berikut :

- a. Memastikan kompatibilitas sistem operasi dengan *tools* pengujian yang dipilih.
- b. Melakukan instalasi dan konfigurasi *tools* pengujian.
- c. Mengkonfigurasi pengaturan jaringan untuk memungkinkan *tools* berkomunikasi dengan sistem yang diuji.
- d. Menjalankan skenario pengujian sederhana untuk memverifikasi bahwa *tools* dapat mengirimkan beban dan mengumpulkan metrik dengan benar.

5. *Test Execution*

Test Execution merupakan tahapan kritis di mana skenario dan test case yang telah dirancang sebelumnya dijalankan untuk menguji kinerja *website* Disnaker Kabupaten XYZ. Pada fase ini, data aktual dikumpulkan untuk mengevaluasi performa, skalabilitas, dan ketahanan sistem di bawah berbagai kondisi beban. Kegiatan yang dilakukan pada tahap ini adalah menjalankan *test case* sesuai dengan rencana pengujian kemudian melakukan pencatatan dan dokumentasi pengujian dan hasil.

6. *Test Closure*

Test Closure merupakan tahap akhir dalam proses functional testing di mana seluruh aktivitas pengujian disimpulkan, data dianalisis, dan dievaluasi. Tahap ini penting untuk memastikan bahwa pengujian telah dilaksanakan dan mendapatkan hasil yang aktual. Adapun kegiatan yang dilakukan pada tahap ini diantaranya melakukan analisis hasil, menyimpulkan dan mengevaluasi hasil pengujian yang telah dilakukan pada *tools* Jmeter dan

2.3.14 Alat Pengujian Perangkat Lunak (*Software Testing Tools*)

Berikut ini merupakan *tools* yang akan digunakan untuk melakukan pengujian di berbagai karakteristik:

1. Selenium IDE

Selenium IDE merupakan sebuah automation testing tool untuk aplikasi web. Selenium IDE dijalankan secara langsung pada browser dan mendukung pada sebagian besar browser yang sudah ada, seperti firefox, chrome, opera, safari dan lain sebagainya. Selenium IDE bisa dijalankan pada berbagai macam platform, seperti windows, linux maupun macOS. Selenium IDE merupakan *tools* yang melakukan system functional testing. Selenium IDE adalah testing *tools* yang fleksibel karena banyak bahasa pemrograman yang mendukung automation testing dengan Selenium IDE, diantaranya adalah Java, python, ruby dan lain sebagainya. Sedangkan, Selenium IDE (*Integrated Development Environment*) adalah *tools* untuk mengembangkan *test case* selenium IDE. Merupakan plug-in yang dapat digunakan di browser Firefox, mudah digunakan dan merupakan cara yang paling efisien untuk mengembangkan *test cases*. Dengan Selenium IDE, penguji bisa memilih elemen UI dari halaman browser yang sedang ditampilkan, kemudian memilih *command* yang telah disediakan dengan parameter yang telah ditentukan sesuai dengan konteks elemen UI yang dipilih. Selenium IDE melakukan tes dengan merekam (*record test*) dan menjalankan hasil rekaman *testing* (Srivastava et al., 2021).

2. Katalon Recorder

Katalon recorder merupakan alat yang bisa menguji aplikasi berbasis mobile, browser, hingga API. Katalon Recorder tidak mengharuskan penggunanya untuk menguasai bahasa pemrograman karena katalon recorder memiliki user interface yang simpel dan informatif sehingga mudah digunakan bagi pemula hingga yang sudah berpengalaman (Zulianto, A dkk, 2021). Katalon recorder adalah aplikasi open source untuk pengujian secara otomatis yang dikembangkan oleh Katalon LLC. Katalon mendukung 3 platform pengujian secara automasi, yaitu Web testing, API testing, dan Mobile testing. Katalon sudah terintegrasi dengan beberapa teknologi luar seperti qtest, JIRA, kobiton, github, dan lain-lain.

3. JMeter

Apache JMeter adalah perangkat lunak yang dikembangkan oleh Apache Software Foundation untuk melakukan pengujian beban dan menganalisis kinerja aplikasi. Awalnya, JMeter ditujukan untuk menguji aplikasi web, namun sekarang dapat digunakan untuk berbagai jenis pengujian kinerja pada sumber daya statis (seperti file HTML atau gambar) maupun dinamis (seperti database atau layanan web). Sebagai aplikasi open source berbasis Java, JMeter memungkinkan pengguna untuk melakukan simulasi lalu lintas pengguna yang besar guna menguji bagaimana aplikasi merespons di bawah berbagai kondisi beban. Salah satu fitur penting dari JMeter adalah kemampuan pengujian terdistribusi, di mana pengujian dapat dilakukan pada beberapa mesin secara bersamaan untuk mensimulasikan beban yang lebih besar. Ini memungkinkan pengujian skala besar yang lebih mendekati situasi nyata, di mana banyak pengguna mengakses aplikasi secara bersamaan dari berbagai lokasi. Selain itu, JMeter mendukung berbagai protokol dan layanan, termasuk HTTP, HTTPS, SOAP, REST, FTP, dan database melalui JDBC, sehingga pengguna dapat melakukan pengujian kinerja pada berbagai jenis aplikasi. Fitur-fiturnya mencakup pengujian beban untuk *server*, grup utas (thread groups) untuk mengelola pengguna virtual, laporan kinerja yang terperinci, serta kemampuan untuk menghasilkan skrip pengujian yang kompleks (Srivastava et al., 2021).

4. Blazemeter

Blazemeter merupakan *tools* pengujian otomatis yang dapat digunakan untuk menguji performa sebuah perangkat lunak. BlazeMeter adalah platform berbasis cloud yang digunakan untuk melakukan pengujian kinerja (*performance testing*) pada aplikasi, situs web, dan API. BlazeMeter memungkinkan pengguna untuk menjalankan berbagai jenis pengujian, seperti load testing, stress testing, dan functional testing, secara scalable. Platform ini mendukung integrasi dengan Apache JMeter, yang merupakan alat open-source untuk pengujian beban, sehingga memungkinkan pengguna untuk memanfaatkan skrip JMeter di lingkungan BlazeMeter begitupun sebaliknya.

5. Maze

Maze merupakan *tools* yang dapat digunakan untuk melakukan usability testing secara online. Maze Design dapat diintegrasikan dengan prototipe desain dari platform seperti Marvel, Sketch, Figma, dan lainnya. Dalam pengujian ini, pengguna atau peserta tes menyelesaikan skenario tugas secara mandiri, tanpa bantuan dari pihak lain. *Tools* ini juga mensimulasikan interaksi dengan membiarkan peserta mengklik tautan untuk berpindah ke layar berikutnya. *Usability testing* menggunakan Maze Design dilakukan secara *online* untuk memudahkan responden dalam mengaksesnya (Glowdy et al., 2020). Maze Design memiliki alat ukur untuk menguji kegunaan, yang digunakan untuk mengevaluasi validitas dan keakuratan desain yang telah dirancang dan diuji. Dalam *usability testing* dengan Maze Design, ada dua aspek utama yang dianalisis yaitu *usability breakdown* dan *heatmap screen*. Analisis kegunaan menunjukkan tampilan layar, rata-rata waktu yang dihabiskan pada setiap halaman, serta rata-rata kesalahan klik yang terjadi. Sementara itu, *heatmap screen* menggambarkan perilaku pengguna di setiap halaman. Semakin merah warna pada halaman, semakin sering responden mengklik area tersebut. Adapun kriteria penilaian berdasarkan *tools* Maze (maze.co) adalah sebagai berikut:

Tabel 2. Nilai dan *Grade Usability Testing Tools Maze*.

Skala Nilai	Kategori
80 - 100	Tinggi
50 - 80	Sedang
0 - 50	Rendah

6. Useberry

Useberry adalah sebuah platform yang dirancang untuk membantu desainer dan tim produk dalam melakukan penelitian terkait UI/UX langsung pada pengguna nyata, dengan menggunakan prototipe desain sebelum produk dikembangkan lebih lanjut. Dengan platform ini, peneliti dapat membuat studi, kemudian membagikannya kepada audiens yang telah dipilih secara khusus sesuai kebutuhan penelitian. Useberry mendistribusikan studi tersebut melalui tautan unik yang dihasilkan oleh peneliti, memungkinkan peserta untuk mengakses dan memberikan tanggapan. Platform ini kemudian mengumpulkan respons dari peserta dan menyediakan hasil yang terperinci kepada peneliti. Useberry memungkinkan para desainer untuk menguji berbagai elemen desain antarmuka sebelum produk mencapai tahap pengembangan penuh, sehingga memungkinkan perbaikan berdasarkan perilaku dan umpan balik langsung dari pengguna. Ini membantu memastikan bahwa produk yang dikembangkan memenuhi harapan pengguna dan meningkatkan pengalaman pengguna (*user experience*) secara keseluruhan. Useberry dapat memonitor metrik penting seperti klik, waktu yang dihabiskan, dan jalur navigasi yang diambil peserta melalui prototipe, memberikan data yang berguna untuk pengambilan keputusan desain (Pratama et al., 2023).

7. Browserstack

BrowserStack merupakan salah satu platform pengujian lintas-browser berbasis *website*. Platform ini memungkinkan pengujian dan *debugging* situs web secara langsung di ribuan browser dan perangkat seluler nyata untuk memastikan aplikasi berjalan dengan baik. Pengujian aplikasi di berbagai browser, sistem

operasi, dan versi yang berbeda secara bersamaan sangat sulit dilakukan secara manual, dan membeli banyak komputer untuk menginstal sistem operasi yang berbeda bisa sangat mahal. Demikian juga, membeli semua perangkat Android dan iOS yang tersedia di pasaran saat ini tidak praktis. Untuk mengatasi masalah ini, BrowserStack menjadi solusi yang ideal karena dapat mengurangi kerumitan dalam menguji berbagai sistem operasi, browser, dan versi yang berbeda. Ketika melakukan pengujian aplikasi seluler, BrowserStack memungkinkan pengembang untuk menghindari pembelian semua perangkat yang ada di pasaran. Platform ini sangat fleksibel dan dapat digunakan di mana saja dan kapan saja. BrowserStack berfungsi sebagai laboratorium jarak jauh dan bisa digunakan seperti peramban desktop nyata. Tidak diperlukan instalasi atau pengaturan tambahan, karena BrowserStack bisa langsung digunakan dengan mengakses URL dan login ke platform dari perangkat apa pun (Lefterov, 2020).

2.3.15 Dimensi Kualitas *Website* Pressman

Menurut Pressman (2010), dimensi kualitas *website* adalah sebagai berikut:

1. Fungsi

Pengujian fungsi bertujuan untuk mengidentifikasi kesalahan yang menunjukkan ketidakcocokan dengan kebutuhan pengguna. Dalam ISO/IEC 25010, ini termasuk dalam sub kategori kebenaran fungsi, yang merupakan bagian dari karakteristik kelayakan fungsional (*functional suitability*).

2. Kinerja

Menurut Pressman (2010) Pengujian kinerja dilakukan untuk memverifikasi bahwa sistem dapat mengatasi beban ekstrim tanpa mengalami penurunan kinerja yang tidak dapat diterima. Dalam ISO/IEC 25010, ini termasuk dalam karakteristik efisiensi kinerja (*performance efficiency*).

3. Portabilitas

Menurut Pressman (2010) pengujian portabilitas bertujuan untuk menemukan kesalahan yang terkait dengan konfigurasi *host* yang unik, baik *hardware* maupun *software*. Dalam ISO/IEC 25010, portabilitas masuk dalam sub kategori adaptabilitas dibawah karakteristik portabilitas. Adaptabilitas

mengukur sejauh mana sistem dapat beradaptasi dengan efektif dan efisien terhadap berbagai *software*, *hardware* atau lingkungan yang berbeda.

4. Kegunaan

Kegunaan pengujian digunakan untuk mengevaluasi sejauh mana pengguna dapat berinteraksi dengan aplikasi *website* dengan efektif, serta sejauh mana tindakan aplikasi *website* tersebut sesuai dengan harapan pengguna (Pressman, 2010). Dalam ISO/IEC 25010, dimensi kegunaan dikategorikan dalam karakteristik *usability*.

2.3.16 Automation Testing

Pengujian perangkat lunak otomatis (*Automation Testing*) melibatkan pembuatan skrip uji menggunakan bahasa skrip seperti Python, JavaScript, atau *Tool Command Language* (TCL), sehingga kasus uji dapat dieksekusi oleh komputer dengan sedikit kontribusi manusia (Sharma, 2014). Adapun pengertian lain menjelaskan bahwa pengujian perangkat lunak otomatis adalah proses membuat program (skrip uji) yang mensimulasikan langkah-langkah kasus uji manual dalam bahasa pemrograman apapun dengan bantuan alat bantu otomatisasi eksternal (Dudekula, 2011). *Automation testing* memiliki banyak kelebihan dibandingkan dengan *manual testing* yaitu Lebih sedikit investasi dalam sumber daya manusia, proses pengujian dengan waktu yang lebih cepat, menjalankan operasi atau tahapan pengujian yang persis sama saat dengan tahapan pengujian kasus uji pada *test case* sebelum dijalankan, mampu menemukan cacat yang belum tentu dapat ditemukan pada saat melakukan *manual testing*, pengurangan biaya jangka panjang karena adanya *error* atau kerusakan perangkat lunak sehingga membutuhkan perbaikan yang memakan biaya cukup besar.

2.3.17 Test Template

Test case (kasus uji) merupakan pengujian perangkat lunak secara manual sehingga pengujian dilakukan pada setiap fitur atau fungsionalitas pada program yang akan diuji (Rahayu, 2020). *Test case* adalah dokumen atau prosedur yang digunakan dalam pengujian perangkat lunak untuk menguji satu atau beberapa fitur atau fungsi tertentu dari program atau aplikasi yang mendefinisikan langkah-langkah yang harus diikuti, data yang digunakan, dan hasil yang

diharapkan untuk setiap situasi uji yang mungkin terjadi. Tujuannya adalah untuk memvalidasi apakah perangkat lunak berperilaku sesuai dengan yang diharapkan dan memastikan bahwa tidak ada kesalahan atau *bug* yang terdeteksi. Dalam menyusun *test case*, ada beberapa atribut yang diperlukan beserta penjelasannya, antara lain:

- a. *Fitur* : Nama atau deskripsi fitur atau fungsionalitas yang akan diuji.
- b. *Karakteristik* : Sub karakteristik yang diuji.
- c. *Test Case ID*: Identifikasi unik yang mengidentifikasi fitur atau fungsionalitas yang akan diuji.
- d. *Test steps*: Urutan langkah-langkah yang akan dilakukan selama pengujian.
- e. *Expected Result*: Merupakan hasil yang diharapkan dari fitur atau fungsionalitas pada program yang akan diuji sesuai dengan *software requirement*.
- f. *Actual Result*: Merupakan hasil sebenarnya yang terjadi pada fitur atau fungsionalitas pada program yang diuji.
- g. *Status*: Kondisi atau status dari fitur yang diuji, apakah berhasil (*pass*) atau tidak berhasil (*fail*).

Jika hasil pengujian tidak sesuai dengan yang diharapkan (*Expected Result*), maka *test* tersebut akan dinyatakan gagal (*fail*), dan sebaliknya. Setelah memahami dan menentukan atribut yang diperlukan, langkah berikutnya adalah menyusun *test case* dalam format tabel sesuai dengan atribut yang sudah ditetapkan sebelumnya (Hasibuan & Dirgahayu, 2021). Model *test case* dalam bentuk tabel dapat dilihat pada Tabel 2.

Tabel 3. Model *Testcase*.

<i>Test Case ID</i>	<i>Fungsi</i>	<i>Karakteristik</i>	<i>Test Step</i>	<i>Expected Result</i>	<i>Actual Result</i>	<i>Status</i>

Tabel 2 merupakan template atau model *test case* yang akan digunakan untuk melakukan dokumentasi pengujian dan pembuatan *test case* dalam penelitian ini.

2.3.18 Bug

Bug perangkat lunak merupakan suatu kesalahan atau kecacatan yang terjadi pada sebuah perangkat lunak. Pendeteksian adanya *bug* sangat penting dilakukan sehingga dapat diperbaiki oleh pengembang agar perangkat lunak berhasil mencapai keinginan atau tujuannya (Utami, 2021).

2.3.19 User Acceptance Testing (UAT)

Pengujian Penerimaan Pengguna (*User Acceptance Testing* - UAT) adalah tahap evaluasi yang melibatkan pengguna akhir dalam interaksi langsung dengan sistem. Tujuannya adalah memverifikasi apakah fungsi-fungsi yang ada telah sesuai dengan kebutuhan dan tujuan yang diharapkan. UAT menggunakan metode pengujian kotak hitam (*black box*) untuk menilai sistem berdasarkan spesifikasinya. UAT memiliki peran penting dalam membuktikan bahwa sistem dapat memenuhi kebutuhan pengguna. Lebih dari sekadar memenuhi spesifikasi teknis, UAT bertujuan untuk memastikan sistem benar-benar sesuai dengan kebutuhan nyata pengguna dalam konteks penggunaan sehari-hari.

Sebagai fase terakhir dalam proses pengujian perangkat lunak, UAT memiliki signifikansi tinggi. Ini merupakan prosedur krusial yang harus dilakukan sebelum perangkat lunak siap dikembangkan lebih lanjut dan diluncurkan ke pasar. UAT juga dikenal dengan istilah lain seperti pengujian beta, pengujian aplikasi, atau pengujian pengguna akhir. Dalam pelaksanaannya, UAT menguji apakah tugas-tugas yang dilakukan sistem telah sesuai dengan spesifikasi yang ditetapkan. Proses ini memainkan peran vital dalam memastikan kualitas dan kesiapan perangkat lunak sebelum dirilis ke publik (Cimperman, 2006). Pada kasus ini UAT digunakan untuk mengukur sub karakteristik *functional appropriateness*.

2.3.20 Pengujian Instrumen Penelitian

Pengujian pada instrumen penelitian berupa angket yang digunakan dalam penelitian ini. Pengujian ini bertujuan untuk mengetahui tingkat validitas dan reliabilitas dari butir pertanyaan yang diberikan kepada responden dalam angket.

1. Uji Validitas

Uji validitas digunakan untuk mengukur sah atau valid tidaknya suatu kuesioner. Suatu kuesioner dikatakan valid jika pertanyaan pada kuesioner mampu untuk mengungkapkan sesuatu yang akan diukur oleh kuesioner tersebut.

Kriteria pengambilan keputusan adalah:

- a. Jika r hitung $>$ r tabel, maka pernyataan tersebut dinyatakan valid.
- b. Jika r hitung $<$ r tabel, maka pernyataan tersebut tidak valid.
- c. Jika $\text{Sig.}(2\text{-tailed}) < 0,05$ maka pernyataan dinyatakan valid.
- d. Jika $\text{Sig.}(2\text{-tailed}) > 0,05$ maka pernyataan dinyatakan tidak valid.

2. Uji Reliabilitas

Reliabilitas menunjukkan bahwa suatu instrumen dapat dipercaya sebagai alat pengumpulan data yang baik dan memberikan hasil yang konsisten. Setelah kuesioner disusun, kuesioner tersebut diuji coba pada beberapa responden untuk mengukur reliabilitasnya. Uji reliabilitas ini digunakan untuk mengetahui seberapa stabil dan konsisten responden dalam menjawab pertanyaan-pertanyaan yang terkait dengan aspek-aspek variabel tertentu yang disusun dalam kuesioner. Uji reliabilitas dapat dilakukan secara bersama-sama terhadap seluruh butir pertanyaan. Jika nilai Alpha $>$ 0,60 maka reliabel dapat dikatakan reliabel (Sujarweni, 2015).

III. METODE PENELITIAN

3.1 Sumber Data

Sumber data dalam suatu penelitian dibutuhkan untuk mendukung proses penelitian. Data merupakan suatu pengetahuan yang diolah agar memiliki makna sehingga menghasilkan informasi. Data primer pada penelitian ini berupa data mengenai alamat *website*, dokumentasi pengerjaan project dalam bentuk *trello* dan dokumen spesifikasi perangkat lunak *website* Disnaker. Pada penelitian ini terdapat data yang digunakan oleh peneliti yaitu sebagai berikut.

3.1.1 Data Primer

Data primer adalah data yang dikumpulkan langsung oleh peneliti dari sumber aslinya untuk tujuan penelitian yang akan dilakukan. Dalam penelitian ini data primer diperoleh melalui metode observasi dan wawancara.

a. Observasi

Observasi adalah proses pengamatan langsung yang dilakukan oleh peneliti terhadap objek atau subjek yang diteliti. Peneliti mencatat perilaku, interaksi, atau fenomena yang terjadi secara alami tanpa campur tangan dari peneliti. Dalam penelitian ini, observasi dilakukan dengan memperhatikan proses pengembangan perangkat lunak untuk *website*, khususnya pada tahap pengujian perangkat lunak yang dilakukan oleh PT Indotechpren.

b. Wawancara

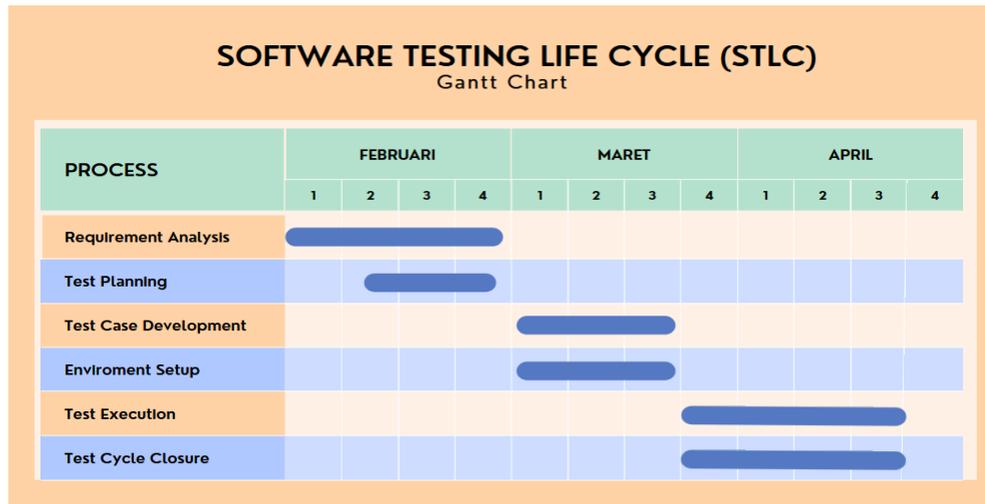
Wawancara adalah proses tanya jawab antara peneliti dan responden atau partisipan penelitian. Tujuan dari wawancara adalah untuk mendapatkan pemahaman yang lebih dalam tentang pengalaman, pandangan, atau persepsi responden terkait topik penelitian. Pada penelitian ini wawancara dilakukan bersama dengan CEO PT Indotechpren sebagai narasumber untuk memperoleh informasi yang lebih rinci terkait pengujian perangkat lunak yang dilakukan pada PT Indotechpren meliputi pihak yang melakukan pengujian, jenis pengujian yang dilakukan, metode pengujian yang digunakan, *tools* yang digunakan untuk melakukan pengujian dan dokumentasi pengujian.

3.1.2 Data Sekunder

Data sekunder adalah data yang sudah ada dan dikumpulkan oleh pihak lain untuk tujuan lain, tetapi kemudian digunakan kembali oleh peneliti untuk penelitian mereka sendiri. Data sekunder diperoleh menggunakan teknik studi literatur dalam penelitian ini berupa hasil penelitian sebelumnya, laporan, statistik, jurnal, *website*, buku, dokumen pemerintah atau data dari sumber elektronik lainnya yang berkaitan dengan topik yang akan diteliti.

3.2 Waktu dan Tempat Penelitian

Penelitian ini dilaksanakan pada Semester Genap Tahun Ajaran 2023/2024 di Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung dengan alamat Jalan Soemantri Brodjonegoro No. 1 Gedung Meneng, Bandar Lampung. Adapun detail waktu pelaksanaan penelitian dapat dilihat pada Gambar 5.



Gambar 5. Waktu Penelitian.

Gambar diatas merupakan ilustrasi dari pembagian waktu dan tahapan yang akan dilakukan pada saat melakukan penelitian. Tahap yang akan dilakukan adalah *Requirement analysis* pada bulan Februari minggu ke 2 hingga 4, *Test Planning* pada bulan Februari minggu ke 2 hingga 4, *Test Case Development* pada bulan Februari minggu ke 2 hingga Maret minggu ke 4, *Environment Setup* pada bulan Februari minggu ke 3 hingga Maret minggu ke 3, *Test Execution* pada bulan Maret minggu ke 4 hingga April minggu ke 3 dan terakhir tahapan *Test Cycle Closure* pada bulan Maret minggu ke 4 hingga April minggu ke 3.

3.3 Alat Penelitian

Dalam penelitian yang dilakukan, akan menggunakan alat penelitian berupa *software* (perangkat lunak) dan *hardware* (perangkat keras) sebagai penunjang.

1. Software (Perangkat Lunak)

Adapun *Software* yang digunakan dalam penelitian ini dibagi berdasarkan karakteristik ISO/IEC 25010 yang akan diuji.

a. Usability

Untuk menguji karakteristik *Usability tools* yang digunakan adalah Maze dan Useberry.

b. *Performance Efficiency*

Untuk menguji karakteristik tersebut *tools* yang digunakan adalah Jmeter dan Blazemeter.

c. *Fungsional Suitability*

Karakteristik *Fungsional Suitability* diuji menggunakan *tools* Selenium IDE dan Katalon Recorder.

d. *Portability*

Karakteristik *Portability* diuji menggunakan *tools* Browserstack.

2. *Hardware* (Perangkat Keras)

Adapun *Hardware* yang digunakan dalam penelitian ini adalah.

- a. *Manufacturer* : Lenovo Ideapad 3 slim 3
- b. *Processor* : AMD Ryzen 3 5300U
- c. RAM : 8 GB
- d. SSD : 512 GB.

3.4 Alur dan Tahapan Penelitian

Penelitian ini menerapkan teknik *Software Testing Life Cycle* (STLC), yang merupakan serangkaian langkah dan proses yang dijalankan dalam pengujian perangkat lunak. STLC mengatur urutan kegiatan yang harus dilakukan dalam siklus pengujian untuk memastikan bahwa perangkat lunak telah diuji secara menyeluruh dan kualitasnya terjamin sebelum dirilis ke pasar atau pengguna akhir. Tahapan tersebut dimulai dari *Requirement Analysis*, *Test Planning*, *Test Case Development*, *Environment Setup*, *Test Execution*, hingga *Test Cycle Closure* sebagaimana diilustrasikan pada Gambar 6.



Gambar 6. *Software Testing Life Cycle* (STLC).

Gambar 6 mengilustrasikan tahapan STLC yang akan menjadi tahapan dari penelitian yang dilakukan. Pada gambar tersebut mengilustrasikan tahapan *Requirement Analysis*, *Test Planning* akan dilakukan pada pembahasan BAB III, sedangkan *Test Case Development*, *Environment Setup*, *Test Execution*, hingga *Test Cycle Closure* akan dilakukan pada pembahasan BAB IV.

3.4.1 Requirement Analysis

Disnaker adalah sebuah platform daring berbasis situs web yang memfokuskan diri pada penyediaan layanan terpadu dalam bidang ketenagakerjaan secara online. Situs ini bertujuan untuk memberikan layanan kepada masyarakat, perusahaan, dan lembaga di Kabupaten tersebut. Dengan menyediakan beragam informasi terkait ketenagakerjaan, pelatihan, sertifikasi, dan aspek lain yang berkaitan di wilayah tersebut. Project ini dikembangkan oleh tim yang terdiri seorang CEO, satu orang *Project Manager* dan *Quality Assurance*, satu orang *Backend Developer* dan satu orang *Frontend Developer*. *Website* ini dikembangkan menggunakan framework dari bahasa pemrograman HTML, CSS, Javascript, PHP. Hasil kegiatan *Requirement Analysis* karakteristik *usability*, *portability*, *functional* dan *performance testing* dijabarkan sebagai berikut :

3.4.1.1 Analisis Proses Bisnis Secara Umum

Dalam era globalisasi dan perkembangan teknologi informasi, Dinas Ketenagakerjaan (Disnaker) memiliki tanggung jawab untuk memastikan bahwa tenaga kerja di wilayahnya memiliki keterampilan dan pengetahuan yang sesuai dengan tuntutan pasar kerja yang terus berubah. Salah satu upaya untuk mencapai tujuan ini adalah melalui penyelenggaraan program pelatihan yang relevan dan efektif dan dari manual menjadi sistem untuk modul pendaftarannya. Berikut adalah beberapa alasan latar belakang *website* Pelatihan Disnaker dengan Modul Registrasi Pelatihan atau Sertifikasi:

a. Transformasi Digital di Bidang Pelatihan

Keberlanjutan dan daya saing tenaga kerja saat ini sangat terkait dengan pemahaman dan penguasaan teknologi. Dengan adanya transformasi digital, *website* pendaftaran informasi pelatihan atau sertifikasi menjadi

sarana yang efisien dan efektif untuk menyebarkan informasi, dan mendaftarkan peserta pelatihan atau sertifikasi secara online.

b. Aksesibilitas dan Keterjangkauan

Membangun *website* pelatihan dengan modul registrasi pelatihan atau sertifikasi memungkinkan aksesibilitas yang lebih luas bagi masyarakat. Peserta pelatihan dapat dengan mudah mendapatkan informasi terkini, mengikuti pelatihan, dan mendaftar dalam pembelajaran tanpa batasan geografis atau waktu.

c. Peningkatan Efisiensi Administrasi

Penggunaan *website* untuk menyelenggarakan pelatihan atau sertifikasi dapat meningkatkan efisiensi administrasi dengan mengotomatiskan proses pendaftaran, pemantauan peserta, dan penyimpanan data. Hal ini dapat mengurangi beban kerja petugas administrasi dan meningkatkan akurasi data.

d. Meningkatkan Keterlibatan Peserta

Website pelatihan dapat menjadi platform interaktif yang memungkinkan peserta terlibat secara aktif dalam proses pelatihan dengan modul pendaftaran pelatihan atau sertifikasi pembelajaran.

e. Peningkatan Transparansi dan Akuntabilitas

Melalui *website*, informasi mengenai program pelatihan, agenda, pengumuman, dan informasi lainnya dapat diakses oleh semua pihak yang terkait. Hal ini meningkatkan transparansi dan akuntabilitas dalam penyelenggaraan pelatihan atau sertifikasi pada modul registrasi pelatihan atau sertifikasi.

f. Peningkatan Reputasi dan Daya Tarik Peserta

Memiliki *website* pendaftaran pelatihan yang informatif dan mudah diakses dapat meningkatkan reputasi Disnaker dan meningkatkan daya tarik peserta dan Mitra Lainnya. *Website* yang profesional dan responsif menciptakan kesan positif terhadap program pelatihan yang diselenggarakan.

g. Kemampuan Analisis dan Pemantauan

Website pelatihan bermodul registrasi pelatihan atau sertifikasi dapat dilengkapi dengan fitur analisis dan pemantauan yang memungkinkan Disnaker untuk mengukur efektivitas program, mengevaluasi tingkat partisipasi, dan merancang pelatihan berdasarkan kebutuhan aktual tenaga kerja dengan menampilkan beberapa grafik.

3.4.1.2 Requirement Analysis Karakteristik Usability.

Kegiatan *requirement analysis* pada karakteristik *usability* menghasilkan luaran sebagai berikut:

1. Tujuan Usability Testing

Tujuan melakukan *usability testing* pada *website* Disnaker Kabupaten XYZ adalah sebagai berikut :

- a. Mengukur dan mengevaluasi seberapa baik kemampuan *website* Disnaker Kabupaten XYZ untuk dikenali oleh pengguna sesuai dengan kebutuhan dan tujuannya. Maze dan Useberry dapat digunakan untuk mengukur seberapa cepat dan akurat pengguna dapat mengenali fungsi atau fitur tertentu dalam *interface* dengan misi "*first-click*" dan melihat seberapa tepat user dapat menentukan tampilan awal yang harus di klik untuk menyelesaikan misi serta waktu yang dibutuhkan pengguna untuk menyelesaikannya. Tujuan ini digunakan untuk mengukur sub karakteristik *appropriateness recognize ability* pada karakter *usability* ISO/IEC 25010.
- b. Mengukur dan mengevaluasi kemampuan *website* Disnaker Kabupaten XYZ untuk dipelajari dengan mudah dan efektif oleh pengguna, serta memberikan kepuasan terhadap tujuan penggunaan tertentu. Maze dan Useberry dapat digunakan untuk melacak bagaimana pengguna mempelajari dan menguasai tugas-tugas baru melihat tingkat keberhasilan partisipan saat menyelesaikan tugas/misi yang diberikan. Tujuan ini digunakan untuk mengukur sub karakteristik *learnability* pada karakter *usability* ISO/IEC 25010.

- c. Mengukur dan mengevaluasi seberapa baik kemampuan *website* Disnaker Kabupaten XYZ untuk dioperasikan dan dikendalikan dengan mudah oleh pengguna. Maze dan Useberry dapat digunakan untuk mengukur kemudahan operasi dengan melacak alur pengguna dalam menyelesaikan tugas dengan menganalisis jumlah klik dan alur navigasi pengguna. Tujuan ini digunakan untuk mengukur sub karakteristik *operability* pada karakter *usability* ISO/IEC 25010.
- d. Mengukur dan mengevaluasi seberapa baik kemampuan *website* Disnaker Kabupaten XYZ untuk melindungi pengguna dari kesalahan yang mungkin dilakukan. Meskipun Maze dan Useberry tidak secara langsung mengukur perlindungan kesalahan, kedua *tools* tersebut dapat mengidentifikasi area di mana pengguna sering membuat kesalahan melalui titik-titik di mana pengguna melakukan tindakan yang tidak diharapkan atau gagal menyelesaikan tugas. Sehingga dapat digunakan sebagai evaluasi oleh tim pengembang. Tujuan ini digunakan untuk mengukur sub karakteristik *user error protection* pada karakter *usability* ISO/IEC 25010.

2. Profil Pengguna

Profil pengguna akan mengilustrasikan karakteristik partisipan secara rinci sehingga dapat digunakan untuk menentukan calon partisipan yang sesuai untuk melakukan pengujian. Adapun karakteristik yang dibutuhkan untuk menjadi partisipan dalam pengujian adalah mereka yang berada pada rentang usia 17 sampai 35 tahun.

3. Sub karakteristik *Usability*

Sub karakteristik yang akan diuji pada *usability testing* ini adalah *Appropriateness recognize ability*, *Learnability*, *Operability* dan *User error protection*.

4. Metrik *Usability Testing*

Metrik yang digunakan dalam *usability testing* adalah sebagai berikut:

V. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil pengukuran yang telah dilakukan terkait dengan Implementasi *Software Product Quality* Berdasarkan Model ISO/IEC 25010 dalam Pengujian *Website* PT. Indonesia Teknologi Preneur (Studi Kasus: Disnaker Kabupaten XYZ), maka dapat ditarik kesimpulan bahwa website Disnaker Kabupaten XYZ memiliki kualitas yang baik pada aspek portabilitas, kinerja, dan kesesuaian fungsional, meskipun terdapat potensi peningkatan pada aspek *usability* dan kompatibilitas perangkat tertentu.

Simpulan ini ditunjang oleh hasil evaluasi sebagai berikut.

- a. Karakteristik *Usability* memperoleh persentase keberhasilan sebesar 89% pada tools Maze dan 87% pada tools Useberry yang dikategorikan tinggi.
- b. Karakteristik *Portability* memperoleh persentase keberhasilan sebesar 99,01% dengan grade A, menandakan kompatibilitas yang sangat baik di berbagai perangkat dan browser. Namun, sistem tidak mendukung Browser Safari versi 13.1 pada macOS Catalina.
- c. Karakteristik *Fungsional Suitability* sebagai berikut.
 - Sub-karakteristik *correctness* menunjukkan 44 dari 50 test case berhasil dijalankan (PASS), dengan tingkat keberhasilan 87,23%, yang mencerminkan sebagian besar fungsi sistem sesuai harapan pengguna.
 - Sub-karakteristik *appropriateness* berdasarkan UAT mencatat nilai 92,5% dengan indeks kepuasan "Sangat Memuaskan". Beberapa fungsi yang dinilai "Baik" atau "Cukup" dapat ditingkatkan.
 - Sub-karakteristik *completeness* memperoleh nilai 0,95, mendekati sempurna (1), menandakan cakupan fungsional yang baik.

- d. Karakteristik *performance efficiency* sistem mencatat waktu respon di bawah 10 detik tanpa error, menunjukkan performa yang sangat baik. Waktu rata-rata eksekusi 136 *milliseconds*, waktu minimum yang dibutuhkan untuk mengeksekusi *thread* yaitu 31 *milliseconds* dan waktu maksimumnya 482 *milliseconds*. *Throughput* menunjukkan bahwa *server* mampu mengeksekusi 0,63 *thread per time unit*.

5.2 Saran

Berdasarkan rangkaian penelitian yang telah diuraikan, ditemukan beberapa aspek yang berpotensi untuk dikembangkan lebih lanjut, di antaranya:

1. Melakukan pengujian terhadap karakteristik lain yang ada pada ISO/IEC 25010, sehingga akan diperoleh hasil pengukuran kualitas perangkat lunak yang lebih komprehensif dan mendukung evaluasi sistem secara optimal.
2. Mempertimbangkan penggunaan perangkat lunak pengujian yang lebih komprehensif, termasuk versi premium atau berlisensi penuh guna meningkatkan kualitas dan keakuratan hasil pengujian. Pendekatan ini berpotensi menghasilkan analisis yang lebih mendalam dan hasil yang lebih optimal dalam proses evaluasi sistem.
3. Melakukan analisis lebih lanjut terkait dengan penyebab kegagalan *portability testing* browser Safari 13.1 pada sistem operasi Mac OS Catalina.
4. Pertimbangan oleh pihak *developer* untuk melakukan perbaikan pada fungsi yang belum sesuai dengan *requirement* yang tercantum dalam Tabel 2 mengenai Dokumentasi Kesalahan Sistem.

DAFTAR PUSTAKA

- Asosiasi Penyelenggara Jasa Internet Indonesia. (n.d.). *Asosiasi Penyelenggara Jasa Internet Indonesia*. Diakses pada 18 November 2024, dari <https://www.apjii.or.id/>.
- British Standards Institution. 2011. BS ISO/IEC 25010:2011. Switzerland: IHS.
- Cimperman, R. (2006). *UAT defined: A guide to practical user acceptance testing*. Pearson Education.
- Dudekula, R.M. 2011. *Automated Software Testing: A Study of the State of Practice*.
- Dhaifullah, I. R., Muttanifudin, M., Salsabila, A. A., & Yakin, M. A. 2022. Survei Teknik Pengujian Software. *JACIS: Journal Automation Computer Information System*, 2(1), 31-38.
- Elgamar. 2020. Buku Ajar Konsep Dasar Pemrograman *Website* Dengan PHP. Malang: Ahlimedia Book.
- G, Myers. 1979. *The Art of Software Testing*. New York: Wiley.
- Ghozali, I. (2006). *Aplikasi analisis multivariate dengan program SPSS* (Edisi ke-4). Semarang: Badan Penerbit Universitas Diponegoro.
- Glowdy, A. G., Fauzi, R., & Alam, E. N. (2020). Perbaikan tampilan user interface untuk meningkatkan user experience pada aplikasi Nganggur.id menggunakan metode user-centered design. *E-Proceeding of Engineering*, 7(2), 7617–7624.
- Habibi, R., & Karnovi, R. 2020. Tutorial Membuat Aplikasi Sistem Monitoring Terhadap Job Desk Operational Human Capital. Bandung: Kreatif Industri Nusantara.
- Huda, M. 2020. *Website* Sebagai Media Informasi Dan Bisnis. *Journal of Community Service and Empowerment*, 1(1).

- Herlinda, H., Katarina, D., Ambarsari, E. W., & Kom, M. 2019. *Automation Testing Tool* Dalam Pengujian Aplikasi Belajar Tajwid Pada Platform Android. *String* (Satuan Tulisan Riset Dan Inovasi Teknologi), 4(2).
- Intyana, S. (2019). *Implementasi ISO/IEC 25010 untuk mengevaluasi website SMA negeri di Kabupaten Pringsewu* (Skripsi, Universitas Lampung).
- Jiang, F., & Lu, Y. 2012. "Software Testing Model Selection Research Based On Yin-Yang Testing Theory," In *Ieee Proceedings Of The International Conference On Computer Science And Information Processing (Cisp)*, Pp. 590-594.
- Jimmy, Felix Utama, Felix, & Albert Prima Laia. 2021. Aplikasi Pembelajaran Penyortiran Menggunakan Algoritma *Super Sort* Berbasis Mobile. *Jurnal SIFO Mikroskil*, 22(1).
- Lefterov, D., & Enkov, S. (2020). *Automated testing framework with browserstack integration. Industry 4.0*, 5(3), 105-108.
- Pranatawijaya, V. H., Widiatry, W., Priskila, R., & Putra, P. B. A. A. (2019). Penerapan skala Likert dan skala dikotomi pada kuesioner online. *Jurnal Sains dan Informatika*, 5(2), 128–137. <https://doi.org/10.34128/jsi.v5i2.185>.
- Pratama, M. B., Carudin, C., & Yusup, D. (2023). Analisis dan Perancangan Ulang User Interface Aplikasi MPP Kota Bogor Menggunakan Metode Design Thinking. *Jurnal Pendidikan Tambusai*, 7(2), 18848-18862.
- Pressman, R. S. 2010. *Software Engineering: A Practitioner's Approach*. New York: McGraw-Hill.
- Pressman, R. S. 2014. *A practitioner's approach. Software Engineering, Eight Edition*.
- Putra, B. R. (2021). *Pengaplikasian ISO/IEC 25010 untuk mengevaluasi website SMKN 1 Palangkaraya* (Skripsi, STMIK Palangkaraya).
- Rahayu, D. K. P. 2020. *Unit Testing Pada Aplikasi Web Mobile (Studi Kasus BisnisJasaLaundry)*.Skripsi.<https://dspace.uui.ac.id/handle/123456789/2834>.
- Riyadi, A. S., Retnandi, E., & Asep. 2012. Perancangan Sistem Informasi Berbasis *Website* Subsistem Guru Di Sekolah Pesantren Persatuan Islam 99 Rancabango. *Sistem Informasi Website*, 09(4), 1–11. <https://doi.org/10.1017/Cbo9781107415324.004>.

- Riyadi, N. R. (2019). Pengujian usability untuk meningkatkan antarmuka aplikasi mobile Myumm Students. *SISTEMASI*, 8(1), 226. <https://doi.org/10.32520/stmsi.v8i1.346>.
- Rosa, A. S., & Shalahuddin. 2014. *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*. Bandung: Informatika.
- Salonen, V. (2012). *Automatic Portability Testing*. Master's Thesis in Information Technology. University of Jyvaskyla.
- Sharma, R M. 2014. *Quantitative Analysis of Automation and Manual Testing*. *International Journal of Engineering and Innovative Technology (IJEIT)* 4(1): 252–57.
- Srivastava, N., Kumar, U., Singh, P. (2021). Software and Performance Testing Tools. *Journal of Informatics Electrical and Electronics Engineering (JIEEE)*, 2(1), 1–12. <https://doi.org/10.54060/JIEEE/002.01.001>.
- Tangkudung, I. (2019). *Evaluasi website menggunakan metode ISO/IEC 25010* (Skripsi, Universitas Negeri Gorontalo).
- Uminingsih, M., Ichsanudin, M. N., Yusuf, M., & Suraya. 2022. Pengujian Fungsional Perangkat Lunak Sistem Informasi Perpustakaan Dengan Metode *Black Box Testing* Bagi Pemula. *STORAGE – Jurnal Ilmiah Teknik dan Ilmu Komputer*, 1(2), 1–8. <https://doi.org/10.55123>.
- Utami, T. W. 2021. *Sistem Informasi Pencatatan Bug Berbasis Web*. Skripsi: Universitas Islam Indonesia.
- Sujarweni, V. W. (2015). *Metodologi penelitian bisnis & ekonomi*. Yogyakarta: Pustaka Baru Press.
- Wicaksono, S. R. (2023). *USABILITY TESTING* (1st ed.). Malang: CV. Seribu Bintang.
- Yuniasri, D., Damayanti, P., & Rochimah, S. (2020). Performance efficiency evaluation frameworks based on ISO 25010. *2020 10th Electrical Power, Electronics, Communications, Controls and Informatics Seminar (EECCIS)*. <https://doi.org/10.1109/EECCIS49483.2020.926343>
- Zulianto, A., Purbasari, A., Suryani, N., Susanti, A. I., Rinawan, F. R., dan Purnama, W. G. (2021). Pemanfaatan Katalon Studio untuk Otomatisasi Pengujian Black-Box pada Aplikasi iPosyandu. *Jurnal Edukasi dan Penelitian Informatika (JEPIN)*, 7(3), 370. <https://doi.org/10.26418/jp.v7i3.46954>.