

# BAB 1

## PENDAHULUAN

---

### 1.1 Latar Belakang

*UI testing* dalam konteks *Quality Assurance* merupakan proses yang esensial dalam pengembangan perangkat lunak. Pengujian ini memastikan bahwa setiap komponen *UI* berfungsi sebagaimana mestinya dan memenuhi spesifikasi yang telah ditetapkan. *UI testing*, atau pengujian antarmuka pengguna, adalah proses pengujian elemen-elemen visual dan interaktif aplikasi untuk memastikan fungsionalitas, tampilan, dan interaksi sesuai dengan spesifikasi dan memberikan pengalaman optimal kepada pengguna akhir. *UI testing* penting dilakukan karena beberapa alasan:

1. Menjamin kepatuhan terhadap spesifikasi : *UI testing* membantu memastikan bahwa setiap elemen *UI* berfungsi sesuai dengan spesifikasi yang telah ditetapkan[1].
2. Mengidentifikasi dan memperbaiki kesalahan dini : Dengan melakukan *UI testing* sejak awal, kesalahan atau bug dapat diidentifikasi dan diperbaiki sebelum aplikasi diluncurkan. Hal ini mengurangi resiko kegagalan aplikasi yang dapat memengaruhi pengalaman pengguna dan reputasi Perusahaan[2]
3. Meningkatkan efisiensi pengembangan : *UI testing* yang efektif membantu meningkatkan efisiensi proses pengembangan. Dengan mendeteksi kesalahan lebih awal, tim pengembangan dapat memperbaiki masalah dengan cepat, mengurangi waktu dan biaya yang dibutuhkan untuk perbaikan di tahap akhir pengembangan[3]
4. Menjaga konsistensi dan keandalan aplikasi : *UI testing* memastikan bahwa aplikasi berfungsi dengan konsisten di berbagai kondisi penggunaan. Ini penting untuk menjaga keandalan aplikasi dan memastikan pengguna mendapatkan pengalaman yang konsisten[4]

Selama periode magang di PT Neuronworks Indonesia, kesempatan diberikan untuk terlibat dalam proses *Quality Assurance* untuk *UI testing* pada aplikasi web yang dikembangkan oleh Perusahaan. Perusahaan ini menyadari pentingnya *UI testing* dalam *quality assurance* untuk memastikan produk akhir yang berkualitas sesuai spesifikasi yang disepakati dengan *client*. Proses ini tidak hanya membantu mendeteksi dan memperbaiki

kesalahan pada tahap awal pengembangan, tetapi juga mencegah masalah yang dapat memengaruhi fungsionalitas aplikasi di masa mendatang. Implementasi *UI testing* yang efektif meningkatkan efisiensi pengembangan dan mengurangi biaya perbaikan dengan memastikan bahwa aplikasi memenuhi standar kualitas yang disepakati[5]

## 1.2 Rumusan Masalah

Dalam konteks *UI testing* beberapa permasalahan muncul yang memerlukan perhatian khusus:

1. Bagaimana cara mengidentifikasi *bug* yang muncul pada *UI* untuk meningkatkan kualitas aplikasi sesuai pedoman *test case*?
2. Bagaimana mendokumentasikan *bug* atau kesalahan pada *UI* aplikasi?
3. Bagaimana memastikan bahwa produk yang dirilis memiliki kualitas sesuai standar pedoman testcase melalui *UI testing* yang efektif?

## 1.3 Tujuan

Quality Assurance ini bertujuan untuk:

1. Teridentifikasinya bug pada UI.
2. Bug pada aplikasi yang terdokumentasi.
3. Produk yang teruji sesuai standar pedoman testcase.

## 1.4 Batasan Masalah

Laporan ini memiliki batasan sebagai berikut:

1. Fokus hanya pada pengujian antarmuka pengguna (*UI testing*).
2. Pengujian tidak mencakup seperti *backend*, performa dan keamanan aplikasi.
3. Skala implementasi pengujian dilakukan dengan koordinasi bersama System Analyst dan Programmer untuk memastikan bahwa semua elemen *UI* diuji sesuai dengan spesifikasi dan kebutuhan proyek. *Feedback* dan perbaikan dilakukan secara berkala sesuai arahan *System Analyst*.
4. Pengujian dilakukan sesuai dengan testcase yang telah disusun dan diberikan oleh *System Analyst*. Pengujian hanya mencakup *scenario* yang terdapat dalam *test case* tersebut.

5. Dokumentasi hasil pengujian dan pelaporan *bug* dilakukan sesuai dengan standar dan template yang telah ditentukan oleh Perusahaan. Laporan akan diserahkan kepada *System Analyst* dan Divisi RIDS(*Research, Inovation, Development Service*) sesuai prosedur yang berlaku.

### 1.5 Definisi Operasional

Dalam pengembangan perangkat lunak, memastikan kualitas dan fungsionalitas produk akhir sangat penting. Untuk mencapai tujuan ini, berbagai konsep dan alat digunakan secara sistematis. Berikut ini akan dijelaskan tiga istilah kunci yang memainkan peran penting dalam proses ini:

1. *Quality Assurance (QA)*

Quality Assurance (QA) adalah aktivitas sistematis yang memastikan produk perangkat lunak memenuhi standar kualitas yang ditentukan, mencakup perencanaan, pengembangan proses, pengujian, pemantauan dan perbaikan.

2. Cypress

Cypress adalah alat pengujian otomatis berbasis JavaScript untuk aplikasi web, memungkinkan instalasi, penulisan tes, eksekusi debugging dan pelaporan hasil pengujian secara cepat dan efisien.

3. *UI Testing (Pengujian Antarmuka Pengguna)*

UI Testing adalah pengujian yang memastikan elemen antarmuka pengguna berfungsi sesuai spesifikasi, termasuk identifikasi elemen, penulisan skrip pengujian, eksekusi pengujian, validasi hasil dan pelaporan.

### 1.6 Metode Pengerjaan

Metodologi yang digunakan dalam laporan ini adalah metode Waterfall. Metode Waterfall adalah pendekatan linear dalam pengembangan perangkat lunak yang mengatur proses dalam beberapa tahap berurutan: Analisis Kebutuhan, Perencanaan Pengujian, Pengembangan Kasus Uji, Persiapan Lingkungan Uji, Pelaksanaan Pengujian, Pelacakan dan Manajemen Cacat, Penutupan Pengujian, dan Pengujian Pasca Rilis. Setiap tahap harus diselesaikan sepenuhnya sebelum melanjutkan ke tahap berikutnya, menjadikannya sangat terstruktur dan ideal untuk proyek dengan persyaratan yang sudah jelas dan tidak mungkin

berubah. Pada tahap Analisis Kebutuhan, semua persyaratan diidentifikasi dan didokumentasikan. Tahap Perencanaan Pengujian melibatkan penyusunan strategi pengujian dan jadwal. Pengembangan *test case* mencakup pembuatan dan dokumentasi *test case*. Persiapan Lingkungan Uji memastikan lingkungan pengujian siap digunakan. Pelaksanaan Pengujian menjalankan kasus uji sesuai rencana. Pelacakan dan Manajemen Cacat mengelola dan melacak cacat yang ditemukan. Penutupan Pengujian menyelesaikan proses pengujian dan mendokumentasikan hasil. Pengujian Pasca Rilis memastikan produk tetap memenuhi standar kualitas setelah dirilis.

### 1.7 Jadwal Pengerjaan

Berikut adalah rincian jadwal pengerjaan selama magang berlangsung :

Tabel 1-1 Jadwal Pengerjaan Juli - Desember

No	Deskripsi Kerja	Juli				Agustus				September				Oktober				November				Desember			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	Orientasi kantor Bersama HCM, Pengenalan Tim dan penandatanganan pakta integritas																								
2	Pembekalan <i>Quality Assurance</i> Neuron Academy																								
3	Pelaksanaan Pekerjaan																								
4	Penyusunan laporan magang																								

**Tabel 1-2 Jadwal Pengerjaan Januari - Juni**

No	Deskripsi Kerja	Januari				Februari				Maret				April				Mei				Juni
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1
1	Pelaksanaan Pekerjaan																					
2	<i>Sharing Knowledge</i>																					
3	Pengembalian id card dan perpisahan dengan tim																					