

BAB II

TINJAUAN PUSTAKA

2.1. Studi Sebelumnya

Nugraha dan Sianturi [9] melakukan penelitian Software testing menggunakan *Black Box* dengan metode *Equivalence Partition*, dengan harapan dapat meningkatkan visibilitas dan memenuhi persyaratan kualitas pada perangkat lunak Evercoss, dengan hasil pengujian untuk menemukan *error* dalam kategori fungsi, antarmuka, struktur data atau akses *database external*, kinerja, inisialisasi dan terminasi. Hasil pengujian memperlihatkan bahwa tidak ada *error* dari setiap fungsi yang diuji, hasil dari pengujian ini dapat menjadikan dokumentasi dari hasil *test result* pengujian yang telah dilakukan.

Penelitian yang dilakukan oleh Dewi, dkk [13] yaitu untuk melakukan pemeriksaan terhadap suatu aplikasi peminjaman buku agar memastikan apakah aplikasi tersebut sudah sesuai dengan kebutuhan pengguna atau belum. Pengujian yang dilakukan oleh penulis menggunakan metode *Black Box* berbasis *Boundary Value Analysis*, penelitian ini menemukan bahwa adanya beberapa ketidaksesuaian pada proses inputan fitur yang terdapat pada aplikasi tersebut, maka dari itu penelitian ini dapat menjadi celah bagi pengembang untuk dapat memperbaiki aplikasi tersebut agar dapat meningkatkan kualitasnya sebelum digunakan oleh pengguna.

Penelitian lain dilakukan oleh Zulkifli, dkk [14] dengan menggunakan metode pengujian *software black box* untuk menemukan *error* pada software dalam beberapa kategori. Dalam pengujian *Black Box*, diperoleh dataset untuk mengukur tingkat akurasi *real output* dan *predictive output*. Hasil penelitian yang dilakukan penulis yaitu menentukan prediksi tingkat akurasi pengujian software *Black Box* dengan teknik *Equivalence Partition* dengan rata-rata akurasi diatas 80%.

Penelitian yang dilakukan oleh Sholeh, dkk [15] menggunakan metode *Blackbox* dengan *Boundary Value Analysis* serta *Equivalence Partitioning* memiliki tujuan untuk mengevaluasi kemampuan aplikasi serta memastikan aplikasi tersebut memiliki kualitas dan hasil yang sesuai dengan harapan. Hasil yang didapatkan dari penelitian ini adalah bahwa pengembangan pada ukmbantul.com sudah mempertimbangkan keterbatasan *data entry* serta formulir yang digunakan telah divalidasi sesuai dengan batasan yang berlaku. Penelitian lain yang dilakukan oleh Marthasari, dkk dengan judul “Pengujian Website Infotech Menggunakan Teknik *Black-Box Decision Table*” [16]. Penelitian ini dilakukan dengan tujuan untuk menguji fungsionalitas perangkat lunak serta menguji kualitas *website* infotech. Penelitian ini dilakukan dengan menggunakan teknik *Black Box* dengan metode *Decision Table*. Hasil penelitian ini yaitu kedua form yang diuji yaitu *login* dan *submit file* telah memenuhi syarat yang ditentukan tanpa adanya permasalahan sehingga dapat disimpulkan bahwa kualitas *website* infotech sudah baik sesuai dengan harapan.

Tabel 2. 1 Perbandingan Penelitian Terdahulu

No	Penulis	Tahun	Metode	Hasil
1	Nugraha, dan Sianturi [9]	2021	<i>Black Box</i> dengan metode <i>Equivalence Partition</i>	Hasil pengujian memperlihatkan bahwa tidak ada <i>error</i> dari setiap fungsi yang di uji, hasil dari pengujian ini dapat menjadikan dokumentasi dari hasil <i>test result</i> pengujian yang telah dilakukan.
2	Dewi, dkk [13]	2022	Metode <i>Black Box</i> dengan menggunakan <i>Boundary</i>	Terdapat beberapa ketidaksesuaian pada proses inputan fitur yang terdapat pada aplikasi tersebut, maka dari itu penelitian ini dapat

No	Penulis	Tahun	Metode	Hasil
			<i>Value Analysis</i>	menjadi celah bagi pengembang untuk dapat memperbaiki aplikasi tersebut agar dapat meningkatkan kualitasnya sebelum digunakan oleh pengguna .
3	Zulkifli , dkk [14]	2020	<i>Black Box</i> dengan teknik <i>Equivalence Partition</i>	Prediksi tingkat akurasi pengujian software <i>Black Box</i> dengan teknik <i>Equivalence Partition</i> dengan rata-rata akurasi diatas 80%
4	Sholeh, dkk [15]	2020	<i>Black Box</i> dengan metode <i>Boundary Value Analysis</i> dan <i>Equivalence Partition.</i>	Pengembangan ukmbantul.com sudah mempertimbangkan keterbatasan <i>data entry</i> serta formular yang digunakan telah divalidasi sesuai dengan batasan yang berlaku
5	Marthasari, dkk [16]	2022	<i>Black Box</i> dengan metode <i>Decision Table</i>	yaitu kedua form yang diuji yaitu login dan submit file telah memenuhi syarat yang ditentukan tanpa adanya permasalahan sehingga dapat disimpulkan bahwa kualitas website infotech sudah baik sesuai dengan harapan.

2.2. Dasar Teori

2.2.1. *Software Testing*

Software Testing atau Pengujian perangkat lunak merupakan sebuah proses mengevaluasi suatu perangkat lunak, mencari ketidaksesuaian, serta mengevaluasi perangkat lunak untuk memastikan apakah aplikasi sudah memenuhi hasil yang diharapkan [17]. Tujuan lain dilakukannya pengujian perangkat lunak yaitu melakukan validasi dan verifikasi suatu aplikasi dalam kondisi yang terkontrol dengan baik [18]. Pada umumnya, pengujian perangkat lunak dilakukan oleh seorang *tester* yang dimana *tester* memiliki tugas utama yaitu memberikan serta menghasilkan *test-case* serta melaporkan ketika menemukan suatu kesalahan di dalam aplikasi [17].

Software testing terbagi kedalam dua kategori yaitu pengujian manual dan pengujian otomatis, Pengujian manual adalah cara pengujian yang dilakukan manual dengan cara memeriksa dokumen, melaporkan, dan mengevaluasi hasil pengujian. Pengujian otomatis merupakan pengujian dengan menjalankan seluruh pengujian manual dengan lebih cepat serta menghemat waktu dan biaya, pengujian ini dilakukan secara otomatis tanpa adanya campur tangan manusia [19]. Hal ini dilakukan agar aplikasi yang akan dirilis menghasilkan kualitas yang sesuai serta terbebas dari *bug* maupun *error*.

2.2.2. *Black Box Testing*

Black Box testing merupakan salah satu teknik pengujian dimana pengujian akan melihat *input* dan *output* perangkat lunak tanpa perlu mengetahui struktur kode perangkat lunak. Pengujian ini dilakukan pada akhir pengembangan perangkat lunak untuk mengetahui apakah perangkat lunak dapat bekerja dengan baik, dalam pengujian *Black Box* pengujian hanya mengetahui *input* dari suatu sistem serta *output* yang diharapkan atau yang dibutuhkan, yang dapat diartikan bahwa pengujian tidak perlu mengetahui cara kerja atau logika dari dalam sistem [20][21]. *Black Box testing* merupakan

pengujian yang dilakukan untuk melengkapi pengujian *White Box* dengan hasil yang cenderung berbeda dari pengujian *White Box* [13]. Pengujian *Black Box* memiliki beberapa keuntungan seperti: Pengujian lebih Efisien, Penguji dapat melakukan testing dengan sederhana dikarenakan penguji tidak perlu mengetahui tentang logika dari dalam sistem, Penguji tidak perlu mengetahui tentang bahasa pemrograman, Pengujian dilihat dari sudut pandang pengguna (*end user*) [8][21].

2.2.3. Equivalence Partitions

Teknik *Equivalence Partitions* merupakan teknik yang dilakukan dengan cara mengurangi jumlah kasus yang diuji dan dibagi data input menjadi beberapa rentang nilai [22], yang dimana didalamnya memiliki beberapa ragam *input* yang dimana input tersebut akan dibagi menjadi beberapa partisi atau dibagi berdasarkan *input* yang memiliki jenis *output* yang serupa [20] , dengan kata lain *Equivalence Partitions* merupakan metode pengujian dengan cara memecah domain *input* dari sebuah *software* yang diuji kedalam beberapa kelas sehingga dapat memperoleh *test case* dari pengujian yang dilakukan, hasil pengujian ini nantinya berupa *valid* atau *invalid* untuk setiap kondisi inputannya [12][20].

Test Case dirancang dalam serangkaian langkah pengujian suatu software atau aplikasi untuk menguji apakah dapat berfungsi dengan baik dalam setiap kategori yang serupa, secara fungsi maupun sifat dari sistem tersebut. *Test Case* dilakukan dengan tujuan untuk memastikan bahwa sistem atau aplikasi yang diuji dapat memproses setiap kategori data *input* dengan benar sesuai dengan rencana [23].

2.2.4. Quality Assurance

Quality Assurance (QA) merupakan suatu proses yang sangat penting didalam siklus SDLC, tujuan dilakukannya QA yaitu untuk mengevaluasi kualitas

software yang telah dikembangkan. Pada tahap ini, pengujian dilakukan untuk menemukan *bug* atau *error* dalam *software* atau aplikasi. Proses pengujian ini penting untuk memastikan *software* yang dikembangkan telah memenuhi persyaratan atau standar kualitas yang telah ditentukan sebelum dirilis ke publik [24]. Dengan melakukan proses QA secara efektif, perusahaan dapat mengurangi resiko kegagalan *software* dan meningkatkan kepercayaan pelanggan terhadap produk yang mereka gunakan. Oleh karena itu, hal ini penting bagi perusahaan untuk memastikan bahwa proses QA dilakukan dengan baik dan efektif dalam setiap siklus SDLC. [25]

2.2.5. Virtual Reality

Virtual Reality (VR) merupakan teknologi yang memungkinkan pengguna untuk dapat memasuki dunia virtual, dunia virtual ini dapat berupa dunia buatan atau sebuah replika dari dunia nyata [26]. Dengan menggunakan VR, pengguna dapat berinteraksi dengan lingkungan yang ada di dalam dunia virtual tersebut layaknya berinteraksi di dunia nyata [27]. Pengalaman ini dapat terjadi dikarenakan VR menggunakan perangkat khusus input dan output seperti *VR Headset*, sensor gerak, dan *controller* yang dapat digunakan oleh pengguna untuk berinteraksi di dalam dunia virtual kapanpun dan dimanapun [26].