

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Judul	Penulis (Tahun)	Metode	Hasil
Pengembangan Media Pembelajaran Game Edukasi Berbasis Android Pada Mata Pelajaran Fisika Dengan Model Prototype	Khoirul Wafiq, Harunur Rosyid, Ayunda Kristari, Farid Sukmana. (2021)	Prototype	Penelitian ini mengembangkan media pembelajaran game edukasi berbasis android menggunakan metode <i>prototype</i> . Metode dimulai dari mendengarkan pelanggan, merancang dan membuat prototype, dan uji coba. Hasil pengujian menunjukkan tingkat usability sebesar 81%, uji fungsionalitas kategori materi 98%, dan kategori uji media 100%, serta performa yang efisien [7].
Penerapan Metode Prototype dalam Pengembangan Sistem Informasi Kuesioner untuk Layanan Survei Terintegrasi	Rhoedy Setiawan*, Zainur Romadhon, Alvin Rainaldy Hakim (2025)	Prototype	Metode <i>prototype</i> digunakan dalam penelitian ini untuk mengembangkan sistem informasi kuesioner secara bertahap melalui siklus interaktif antara pengembang dan pengguna. Proses dimulai dari komunikasi untuk menggali

			<p>kebutuhan, kemudian dibuat rancangan awal sistem (quick design) yang langsung diwujudkan menjadi prototype sederhana. Prototype tersebut diuji dan dievaluasi oleh pengguna, lalu dikembangkan kembali dalam siklus berikutnya hingga sistem memiliki fitur yang lebih lengkap. Dengan pendekatan ini, sistem dapat disesuaikan secara cepat dengan kebutuhan nyata pengguna dan menghasilkan aplikasi yang lebih efektif serta efisien.</p>
<p>Rancang Bangun Sistem Inventory Pada Izara Batik Menggunakan Metode Prototype</p>	<p>Nur Hayatia, M.Rudi Fanajib, Mujibul Hakimc (2025)</p>	<p>Prototype</p>	<p>Penelitian ini merancang dan membangun sistem inventory pada Izara Batik menggunakan metode <i>prototype</i> untuk mengelola serta memantau persediaan bahan baku maupun produk jadi secara lebih efektif dan efisien. Sistem dikembangkan melalui tahapan komunikasi, perancangan cepat, pembuatan <i>prototype</i>,</p>

			<p>hingga uji coba menggunakan blackbox testing yang menunjukkan semua fungsi berjalan sesuai harapan.</p>
<p>Aplikasi Berbasis Android Menggunakan Flutter Framework Untuk Keperluan Perizinan Tugas Keluar Pada PT. XYZ</p>	<p>Indhi Rahmawati*, Dian Permata Sari (2024)</p>	<p>Prototype</p>	<p>Penelitian ini menghasilkan aplikasi Android bernama Penelitian ini menggunakan metode <i>prototype</i> untuk membangun sistem CarOut untuk mempermudah karyawan dalam mengajukan perizinan tugas keluar. Tahapan dimulai dengan analisis kebutuhan, pengembangan sistem, pengujian dan evaluasi sistem menggunakan blackbox dan <i>Technology Acceptance Model</i>, revisi dan penyempurnaan sistem. Aplikasi ini mampu meningkatkan efektivitas dan efisiensi proses perizinan tugas keluar, sekaligus mengurangi penggunaan kertas dan hambatan persetujuan manual [8].</p>

Tabel 2. 1 Penelitian terdahulu

2.2 *Quality Assurance (QA)*

Quality Assurance (QA) atau Jaminan Mutu merupakan bagian dari manajemen mutu yang berfokus pada upaya untuk memberikan keyakinan bahwa seluruh persyaratan mutu dapat terpenuhi [9]. Secara teknis, jaminan mutu pengujian dapat dipahami sebagai rangkaian kegiatan yang sistematis dan terencana dalam proses pengujian, dengan tujuan memberikan keyakinan yang cukup bahwa data yang dihasilkan sesuai dengan standar mutu dan dapat diterima oleh pengguna [10].

2.3 *Ujian Validasi Kopi*

Ujian validasi kopi adalah ujian yang dirancang untuk mengukur dan mengevaluasi kemampuan peserta ujian dalam mengidentifikasi, membedakan, dan mengukur intensitas rasa, baik itu produk maupun sampel rasa. Proses pengujian ini memiliki peranan penting dalam menjaga mutu, karena melalui ujian tersebut dapat dipastikan bahwa setiap produk kopi yang dihasilkan memiliki standar kualitas yang konsisten dan tetap terjaga dari waktu ke waktu.

Saat ini, divisi Quality Assurance (QA) di PT XYZ merasa tidak efektif dalam melakukan tugasnya, terutama pada tahap persiapan pelaksanaan uji validasi kopi. Proses pembuatan soal ujian validasi masih dilakukan secara manual dengan memanfaatkan program Microsoft Excel. Pada tahap ini, divisi QA dituntut untuk membuat sejumlah kode acak yang dilengkapi dengan nilai untuk setiap kode yang dihasilkan. Jumlah kode yang harus disiapkan tidak sedikit, yakni berkisar antara 13 hingga 27 kode untuk setiap peserta, bergantung pada jenis ujian yang diikuti. Setiap kode juga harus bersifat unik, jadi tidak boleh ada duplikasi, yang membuat proses pembuatan menjadi semakin rumit. Kompleksitas tersebut bertambah ketika dalam satu waktu terdapat lebih dari satu jenis ujian yang harus dilaksanakan secara bersamaan, sehingga menambah beban kerja bagi divisi QA. Kondisi ini tidak hanya mengurangi efektivitas, tetapi juga meningkatkan potensi terjadinya kesalahan, seperti adanya kode ganda, nilai yang tidak sesuai, atau ketidaksesuaian antara soal dan jenis ujian. Kesalahan-kesalahan tersebut pada akhirnya dapat berpengaruh buruk terhadap pelaksanaan ujian validasi kopi perusahaan.

Proses pembuatan soal ujian dimulai dengan menghitung jumlah total kode yang dibutuhkan, yaitu hasil perkalian antara jumlah peserta ujian yang dimasukkan dengan jumlah kode jenis ujian yang dipilih. Setiap jenis ujian memiliki daftar nama ujian masing-masing. Untuk menjamin keunikan kode, digunakan objek Set pada bahasa pemrograman TypeScript yang mampu menyimpan kumpulan nilai unik. Kode yang dihasilkan berupa 4 digit angka dalam rentang 1000-9999, yang diperoleh melalui fungsi `Math.random()`. Prosesnya dilakukan dengan perulangan hingga jumlah kode unik dalam Set sama dengan total kode yang dibutuhkan. Setelah terkumpul, kode-kode tersebut dibagi sesuai kebutuhan masing-masing jenis ujian, lalu dipasangkan dengan nilai ujian terkait.

2.4 Prototype

Secara umum, *prototype* dapat dipahami sebagai rancangan awal dari suatu sistem yang berfungsi membentuk model serta memberikan gambaran ukuran dan skala yang akan dikembangkan pada tahap berikutnya [11]. *Prototype* dapat diwujudkan dalam berbagai tingkat, mulai dari *prototype* sederhana (low-fidelity) yang hanya berupa sketsa atau diagram konsep, hingga *prototype* yang lebih kompleks (high-fidelity) dengan detail menyerupai sistem akhir yang diharapkan [12]. Melalui penggunaan *prototype* ini, baik pengembang maupun pengguna dapat berinteraksi langsung dengan rancangan tersebut tanpa harus membuat produk fisik yang sesungguhnya.

2.5 Black Box Testing

Black box testing merupakan metode pengujian perangkat lunak yang berfokus pada fungsi sistem tanpa memperhatikan detail internal maupun struktur kode program di dalamnya [13]. Black box testing bertujuan utama untuk memastikan setiap fungsi dalam aplikasi berjalan dengan benar sesuai kebutuhan pengguna serta spesifikasi sistem yang telah ditetapkan. Metode pengujian ini efektif dalam mengidentifikasi kesalahan pada alur logika bisnis, tampilan antarmuka, maupun proses validasi data. Pengujian diawali dengan menentukan setiap kebutuhan fungsional sistem, kemudian membuat skenario pengujian setiap fungsionalitas, dan hasil yang diharapkan.

2.6 User Acceptance Testing

User Acceptance Test (UAT) adalah suatu bentuk pengujian perangkat lunak yang dilakukan oleh pengguna akhir untuk memastikan bahwa sistem yang telah dikembangkan telah sesuai dengan kebutuhan dan harapan pengguna. Tujuannya adalah untuk memastikan bahwa sistem sesuai dengan harapan pengguna serta dapat terintegrasi dengan baik ke dalam alur kerja mereka [14].

