BAB II
TINJAUAN PUSTAKA

2.1 Studi Sebelumnya

Beberapa penelitian terdahulu yang telah melakukan evaluasi sistem berbasis web maupun aplikasi mobile menggunakan metode kualitatif, menurut Mazaheri Habibi dkk. [6] melakukan penelitian yang bertujuan mengevaluasi kegunaan (*usability*) sistem manajemen informasi tempat tidur rumah sakit (*Bed Information Management System*/BIMS) guna mengidentifikasi hambatan dalam interaksi manusia dan komputer. Penelitian ini menggunakan metode *Think Aloud Protocol* yang diterapkan pada 16 pengguna di 50 rumah sakit non-pendidikan di Iran. Hasilnya ditemukan 80 permasalahan, dengan permasalahan entri data (48%) dan tata letak (41%) sebagai yang paling banyak ditemukan. Penelitian ini menyimpulkan bahwa evaluasi usability perlu dilakukan sebelum sistem diimplementasikan agar kesalahan dalam proses penanganan pasien dapat diminimalkan

Wijaya & Perdana[7] melakukan penelitian yang bertujuan membangun ruang partisipasi masyarakat desa melalui sistem *Focus Group Discussion* (FGD) berbasis website. Latar belakang penelitian ini adalah rendahnya partisipasi masyarakat dalam diskusi pembangunan desa akibat keterbatasan informasi dan akses. Dengan menggunakan metode pengembangan perangkat lunak *waterfall*, penelitian ini menghasilkan aplikasi FGD berbasis website yang memungkinkan pemerintah desa dan masyarakat berdiskusi secara daring, kapan saja dan di mana saja, sehingga partisipasi masyarakat dapat menjadi lebih menyeluruh.

Riswandi & Arnova [8]melakukan penelitian untuk menganalisis pengaruh penerapan sistem informasi akuntansi, khususnya platform Handi HTS dan Web Portal, terhadap kinerja karyawan PT. Ajinomoto Cabang Bengkulu. Penelitian ini menggunakan pendekatan kualitatif

deskriptif dengan teknik pengumpulan data melalui wawancara mendalam, observasi, dan *Focus Group Discussion* (FGD) terhadap 15 informan dari berbagai jabatan. Hasil penelitian menunjukkan bahwa kedua sistem memberikan dampak positif yang signifikan terhadap lima dimensi kinerja karyawan, yaitu efisiensi waktu, kualitas kerja, produktivitas, kompetensi, dan kepuasan kerja, melalui penyediaan informasi secara *real-time* dan pelaporan otomatis.

Pratama dkk. [9]melakukan penelitian yang bertujuan mengevaluasi *usability* aplikasi mobile Padiciti, sebuah platform *m-ticketing*, dari dua perspektif sekaligus, yaitu perspektif pengguna dan perspektif ahli. Penelitian ini mengombinasikan metode *Think Aloud Protocol* dengan 10 pengguna dan *Heuristic Evaluation* dengan 4 evaluator ahli. Dari metode *Think Aloud* ditemukan 145 permasalahan yang dikelompokkan menjadi 35 permasalahan unik, sementara *Heuristic Evaluation* menghasilkan 64 permasalahan yang dikelompokkan menjadi 30 permasalahan unik. Secara keseluruhan, terdapat 40 permasalahan gabungan yang dikategorikan ke dalam tiga aspek: fungsional, tampilan, dan informasi, beserta rekomendasi perbaikannya.

Wibisono dkk.[10] melakukan penelitian yang bertujuan mengevaluasi *usability* aplikasi mobile Sipindo, sebuah sistem informasi pertanian berbasis Android. Serupa dengan penelitian sebelumnya, penelitian ini juga menggabungkan metode *Think Aloud* dengan 10 pengguna (petani dan mahasiswa pertanian) serta *Heuristic Evaluation* dengan 3 evaluator ahli. Dari *Heuristic Evaluation* ditemukan 46 permasalahan, dengan permasalahan terbanyak pada prinsip *match between system and the real world*. Sementara dari metode *Think Aloud* ditemukan 55 permasalahan, dengan tugas login sebagai yang paling bermasalah. Penelitian ini menghasilkan rekomendasi perbaikan yang mencakup aspek fungsional, tampilan, dan informasi, termasuk perbaikan alur *login* Facebook, pembaruan harga produk, serta kelengkapan detail toko.

Berdasarkan studi sebelumnya, sebagian besar penelitian berfokus pada evaluasi usability aplikasi yang telah tersedia maupun pengembangan aplikasi berbasis web. Sementara itu, penelitian ini berfokus pada perancangan solusi stok opname digital berbasis Google Workspace yang disesuaikan dengan proses bisnis perusahaan serta dievaluasi dari aspek usability menggunakan Think Aloud Protocol.

Tabel 2.1 Studi Sebelumnya

No	Penulis	Tahun	Tujuan	Metode	Hasil
1.	Mohammad dkk[6]	2018	Mengevaluasi kegunaan (*usability*) sistem manajemen informasi tempat tidur (BIMS) untuk mengidentifikasi hambatan interaksi manusia dan komputer	Think Aloud Protocol	Terdapat masalah pada 80 entri data/tata letak tampilan sistem manajemen tempat tidur.
2.	Masesa Angga Wijaya dan Chepy Perdana [7]	2023	Membangun ruang partisipasi masyarakat desa melalui sistem *Focus Group Discussion* (FGD) berbasis website untuk mempermudah penyampaian aspirasi	FGD	Terciptanya aplikasi FGD berbasis website yang memungkinkan pemerintah desa dan masyarakat berdiskusi secara transparan kapan saja dan di mana saja
3.	Herdyasnyah dkk [8]	2025	Menganalisis pengaruh sistem informasi akuntansi (Handi HTS dan Web Portal) terhadap kinerja karyawan di PT. Ajinomoto Cabang Bengkulu.	FGD	Penerapan sistem meningkatkan efisiensi waktu, kualitas kerja, produktivitas, kompetensi, dan kepuasan kerja melalui penyediaan informasi *real-time*.

4.	Pratama dkk. [9]	2019	Mengevaluasi usability aplikasi mobile Padiciti (m-ticketing) dari perspektif pengguna dan ahli untuk meningkatkan daya tarik pengguna	Think Aloud Protocol	Ditemukan 40 permasalahan rekomendasi perbaikan pada aspek fungsional, tampilan, dan informasi.
5.	Wibisono dkk. [10]	2023	Mengevaluasi usability aplikasi mobile Sipindo (sistem informasi pertanian) untuk menemukan permasalahan dan memberikan rekomendasi perbaikan.	Think Aloud Protocol	Ditemukan 55 permasalahan rekomendasi perbaikan fungsional, tampilan, dan informasi.

2.2. Dasar Teori

2.2.1 Stok Opname

Stok opname adalah kegiatan pendataan dan pemeriksaan barang perusahaan secara menyeluruh melalui penghitungan kesesuaian antara jumlah barang pada data sistem dengan kondisi fisik aktual di lapangan. Kegiatan ini bertujuan untuk mengetahui secara pasti apakah nilai persediaan barang dagang benar-benar sama atau terdapat selisih kelebihan maupun kekurangan persediaan, sekaligus memantau aset guna memelihara kualitas koleksi perusahaan[11].

2.2.2 Google Work Space

Google Workspace adalah rangkaian aplikasi produktivitas berbasis *cloud* yang dikembangkan oleh Google, yang mencakup berbagai layanan saling terintegrasi seperti Google Form, Google Sheets, dan Google Apps Script. Dalam penelitian ini, ketiga layanan ini memiliki fungsi dan sistem kerjannya tersendiri sebagai berikut:

2.2.2.1 Google Form

Google Form adalah aplikasi berbasis web gratis dari Google Workspace yang digunakan untuk membuat formulir online seperti survei, kuesioner, dan pendaftaran, dengan data yang otomatis tersimpan dan dapat diolah secara real-time melalui integrasi dengan Google Sheets [12]. Dalam penelitian ini, Google Form digunakan sebagai antarmuka input data stok opname, sehingga pengguna dapat mencatat data secara langsung tanpa perlu membangun aplikasi input khusus.

2.2.2.2. Google Sheet

Google Sheets adalah aplikasi pengolah angka (*spreadsheet*) berbasis *cloud* dari Google Workspace yang memungkinkan pengeditan data secara real-time, penyimpanan otomatis, serta pengolahan data melalui formula dan fungsi bawaan. Google Sheets dapat diintegrasikan dengan produk perangkat lunak Google lainnya, seperti Google Docs, Gemini, dan lain-lain, serta mampu digunakan oleh beberapa pengguna sekaligus sehingga memungkinkan pengguna untuk berkolaborasi dalam bekerja [13]. Dalam penelitian ini, Google Sheets tidak hanya berfungsi sebagai *spreadsheet*, tetapi juga sebagai basis data sederhana yang digunakan untuk menyimpan, memproses, dan menghasilkan laporan secara otomatis. Data yang telah dikumpulkan dan disimpan di Google Sheets dapat diekspor, namun memerlukan sedikit penyesuaian tabel agar sesuai dengan format tabel sistem POS.

2.2.2.3 Google Apps Script

Google Apps Script adalah platform pengembangan berbasis *cloud* yang digunakan untuk mengotomatisasi dan memperluas fungsi layanan Google Workspace, seperti Google Sheets, Google Form, dan Gmail. Dalam penelitian ini, Google Apps Script berperan sebagai penghubung antara antarmuka input, basis data, dan logika pemrosesan, sehingga memungkinkan terciptanya sistem informasi sederhana tanpa memerlukan server khusus [14].

2.2.3 Usability ISO 9241-11

Usability merupakan tingkat sejauh mana suatu sistem dapat digunakan oleh pengguna tertentu untuk mencapai tujuan tertentu secara efektif, efisien, dan memberikan kepuasan dalam konteks penggunaan tertentu. Menurut standar ISO 9241-11, usability didefinisikan menjadi tiga aspek, yaitu *Effectiveness* (kemampuan pengguna untuk mencapai

tujuan mereka dalam konteks yang spesifik), *Efficiency* (sumber daya yang digunakan terkait dengan akurasi dan kesempurnaan yang dicapai pengguna dalam menjalankan tugas), dan *Satisfaction* (tingkat kepuasan pengguna saat menggunakan aplikasi). [15]. Ketiga aspek tersebut saling berkaitan dan digunakan untuk mengevaluasi apakah suatu sistem dapat diterima serta mudah digunakan oleh penggunanya[16].

Aspek *effectiveness* berkaitan dengan pencapaian tujuan tertentu dengan kesesuaian kondisi yang dilakukan berdasarkan kemampuan pengguna, yang secara umum diukur saat pengguna menyelesaikan tugasnya. Aspek *efficiency* berkaitan dengan pencapaian tujuan tertentu berdasarkan kecepatan dan akurasi pengguna dalam menggunakan sistem. Sementara itu, aspek *satisfaction* berkaitan dengan tingkat kepuasan dan kenyamanan pengguna selama berinteraksi dengan sistem, yang dapat diperoleh baik melalui data kuantitatif maupun data kualitatif deskriptif berupa perasaan, persepsi, dan verbalisasi pengguna [15].

Pendekatan ini menunjukkan bahwa meskipun definisi konseptual ketiga aspek usability mengacu pada standar ISO 9241-11 yang sama, teknik operasionalisasinya dapat disesuaikan dengan konteks studi kasus dalam hal ini sistem pencatatan stok opname berbasis Google Form dan Google Sheets pada CV XYZ.

2.2.4 *Think Aloud Protocol* (TAP)

Think Aloud Protocol (TAP) adalah metode evaluasi usability di mana responden diminta untuk mengucapkan secara verbal seluruh pikiran, pendapat, dan perasaan yang muncul selama mereka berinteraksi dengan sistem yang sedang diuji. Metode ini pertama kali dipopulerkan oleh Ericsson & Simon dalam bidang psikologi kognitif dan kemudian diadopsi secara luas dalam penelitian usability sistem informasi[10].

Terdapat dua varian utama TAP, yaitu sebagai berikut.

1. *Concurrent Think Aloud* (CTA): Responden mengucapkan pikirannya secara bersamaan (simultan) dengan saat mereka mengerjakan tugas. Metode ini menghasilkan data verbal yang kaya karena mencerminkan proses berpikir secara langsung, namun dapat memengaruhi kecepatan dan alur kerja alami responden[17].
2. *Retrospective Think Aloud* (RTA): Responden mengucapkan pikirannya setelah selesai mengerjakan tugas, biasanya sambil menonton rekaman sesi pengujian mereka sendiri.

Metode ini lebih alami dalam hal performa tugas, tetapi rentan terhadap bias memori [17].

Berdasarkan 2 jenis TAP, dapat disimpulkan bahwa masing masing memiliki fungsi dan nilainya sendiri, dalam penggunaan metode *Concurrent Think Aloud* , para responden saat menjalan *task* mengeluarkan kata kata verbal atas respon terhadap solusi yang digunakan. Beda dengan *Retrospective Think Aloud* metode ini para responden diwajibkan menyelesaikan semua task kemudian setelah selesai menjalan *task*, memberikan *feedback* hasil dari pengujiannya.

2.2.5 *Verbal Protocol Analysis* / VPA

Verbal Protocol Analysis adalah metode analisis kualitatif yang digunakan untuk mengolah data verbal pada pilar efektivitas usability yang diperoleh dari sesi *Concurrent Think Aloud*. Metode ini berfokus pada proses transkripsi dan pengkodean pernyataan verbal responden guna mengidentifikasi pola perilaku, kesulitan, dan pemikiran pengguna selama berinteraksi dengan suatu sistem[18]. Dalam melakukan analisis dengan metode ini peneliti menggunakan acuan *codebook* sebagai landasan dalam pengkodean di transkrip.

Codebook adalah panduan pengkodean yang berisi definisi operasional dari setiap kategori kode yang digunakan dalam proses analisis data kualitatif. Pada penelitian nini *Codebook* berfungsi sebagai acuan bagi peneliti dalam mengklasifikasikan data pada transkrip yang diperoleh dari pengujian menggunakan metode *Concurent Think Aloud*. sehingga proses pengkodean dapat dilakukan secara runtut dan sesuai dengan teori yang digunakan[19].

Dalam konteks evaluasi usability menggunakan Think Aloud Protocol, *codebook* digunakan untuk mengklasifikasikan hasil interaksi responden dengan sistem ke dalam kategori-kategori yang mencerminkan tingkat keberhasilan penyelesaian tugas. Penggunaan *codebook* memungkinkan peneliti untuk mengukur efektivitas sistem secara kuantitatif berdasarkan data kualitatif dari pengujian *Concurent Think Aloud*.

Pada penelitian ini, *codebook* yang digunakan mengadopsi kerangka klasifikasi *task outcome* berdasarkan penelitian *Berry[20]*, dengan 2 kategori sebagai berikut:

Tabel 2.2 *Codebook* dari *slip* dan *mistake*

Code	Definisi
Slip	Responden pada dasarnya memahami cara menggunakan sistem, namun

	melakukan kesalahan kecil yang tidak disengaja dan kemudian segera menyadari serta memperbaikinya sendiri tanpa bantuan
Mistake	Responden melakukan kesalahan yang bersumber dari pemahaman yang keliru terhadap sistem atau antarmuka, sehingga responden memerlukan waktu lebih lama atau melakukan pengulangan langkah untuk menyelesaikan tugas

Kedua kategori ini digunakan dalam proses *Verbal Protocol Analysis* untuk setiap *task* yang diberikan kepada responden, sehingga hasil pengkodean dapat digunakan sebagai dasar perhitungan tingkat efektivitas sistem berdasarkan standar ISO 9241-11.

2.2.6 *Time on Task*

Time on task adalah metode pengukuran kuantitatif yang digunakan untuk mengolah data waktu pada aspek efisiensi usability yang diperoleh dari sesi pengujian *task scenario*. Metode ini berfokus pada pencatatan waktu mulai dan waktu selesai responden dalam mengerjakan setiap tugas guna mengidentifikasi seberapa cepat atau lambat pengguna dalam menyelesaikan interaksinya dengan suatu sistem[21]. Dalam melakukan pengukuran dengan metode ini, peneliti menggunakan data waktu mulai dan waktu selesai sebagai dasar dalam menghitung durasi penyelesaian setiap tugas.

Durasi penyelesaian tugas adalah selisih antara waktu responden mulai mengerjakan suatu tugas hingga responden menyatakan tugas tersebut selesai dikerjakan. Semakin singkat durasi yang dibutuhkan responden untuk menyelesaikan suatu tugas, maka semakin efisien sistem tersebut digunakan [21].

Dalam konteks evaluasi usability, *time on task* pada penelitian ini digunakan untuk membandingkan durasi penyelesaian tugas responden pada kondisi sebelum (proses pencatatan stok opname secara manual) dan kondisi sesudah (proses menggunakan sistem digital berbasis Google Form dan Google Sheets) diterapkannya solusi yang dikembangkan. Perbandingan durasi antara kedua kondisi tersebut kemudian dihitung persentase efisiensi waktunya, sehingga dapat diketahui besarnya peningkatan efisiensi yang dihasilkan oleh solusi yang dikembangkan.

2.2.7 Verbal Sentiment Analysis

Dalam evaluasi *usability* berbasis *Think Aloud Protocol*, tingkat kepuasan pengguna dapat dilihat dari protokol verbal selama interaksi pengujian Google Form dan Google Sheet berlangsung[22]. Pengolahan data kualitatif ujaran responden dilakukan melalui pendekatan *Verbal Sentimen Analysis* secara manual (*human coding*) memanfaatkan *software* QualCoder**.** Pengkodean manual ini mengelompokkan ujaran evaluatif responden ke dalam tiga variabel sentimen utama:

1. Positif (*Satisfied*): Ungkapan verbal yang mengekspresikan kepuasan, kemudahan, kelancaran, atau apresiasi terhadap elemen antarmuka dan alur kerja.
2. Negatif (*Dissatisfied*): Ungkapan verbal yang mengekspresikan kebingungan, keluhan, hambatan visual/navigasi, atau kekecewaan terhadap sistem.
3. Netral (*Acceptable*): Ungkapan verbal yang bersifat evaluatif-prosedural, pernyataan ragu di awal yang berhasil teratasi secara mandiri, maupun penilaian objektif tanpa muatan emosi positif/negatif yang dominan.

Pendekatan pengkodean manual terhadap ujaran verbal ini dipilih, bukan kuesioner baku seperti System Usability Scale (SUS), karena data yang dianalisis diperoleh langsung dari protokol Concurrent Think Aloud yang berjalan bersamaan dengan pengujian task, sehingga mencerminkan reaksi pengguna pada saat (in-the-moment) berinteraksi dengan elemen antarmuka tertentu, bukan penilaian global dan retrospektif seperti yang dihasilkan oleh instrumen kuesioner. Pendekatan ini memungkinkan setiap ujaran sentimen ditautkan langsung ke task dan elemen antarmuka penyebabnya, sebagaimana disajikan pada Tabel 4.9, suatu tingkat kedalaman kontekstual yang secara umum tidak dapat diperoleh dari skor tunggal kuesioner baku. Peneliti menyadari bahwa jumlah ujaran evaluatif yang dianalisis (22 ujaran dari 8 responden) tergolong terbatas, sehingga hasil pengkodean sentimen ini bersifat indikatif dan deskriptif untuk mengidentifikasi pola kecenderungan persepsi verbal pengguna pada solusi yang diuji, dan tidak dimaksudkan untuk digeneralisasikan secara statistik ke populasi pengguna secara keseluruhan.

2.2.8 *Low Code Coding*

Low Code Coding merupakan pendekatan pengembangan perangkat lunak yang

memungkinkan sebuah aplikasi dibangun melalui antarmuka visual berupa komponen siap pakai dan mekanisme *drag-and-drop*, sehingga kebutuhan penulisan kode secara manual dapat diminimalkan tanpa sepenuhnya dihilangkan[23]. Karakteristik utama dari platform low-code mencakup penggunaan *Integrated Development Environment* (IDE) yang bersifat visual untuk menentukan antarmuka dan alur kerja aplikasi, serta *connector* yang menghubungkan sistem *back-end* secara otomatis, meskipun pengembang tetap dapat menambahkan kode secara manual apabila terdapat kebutuhan khusus yang tidak dapat dipenuhi oleh komponen visual yang tersedia maka hal inilah yang membedakan *low-code* dari pendekatan *no-code*, di mana proses pengembangan sepenuhnya bergantung pada antarmuka visual tanpa penulisan kode sama sekali.

Dalam konteks penelitian ini, sistem stok opname yang dikembangkan di CV XYZ merupakan salah satu bentuk penerapan pendekatan low-code, karena dibangun di atas ekosistem Google Workspace dengan memanfaatkan Google Spreadsheet sebagai basis data dan Google Apps Script sebagai lapisan logika, yang keduanya menyediakan komponen visual dan API bawaan sehingga peneliti tidak perlu membangun infrastruktur backend maupun antarmuka pengguna dari awal. Sedikit penulisan kode yang tetap dilakukan pada Apps Script untuk kebutuhan seperti validasi data opname, penentuan indikator status stok secara otomatis (Aman/Hampir Habis/Habis) yang ditampilkan pada Google Form dan sheet Dashboard, hingga fitur ekspor data mencerminkan karakteristik low-code yang tetap memungkinkan intervensi kode terbatas di luar komponen visual yang tersedia, yang pada akhirnya memungkinkan pengembangan sistem dilakukan dengan sumber daya teknis yang minim, sesuai dengan latar belakang penelitian yang menekankan keterbatasan sumber daya IT pada usaha distribusi skala menengah seperti CV XYZ.