

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Peneliti memberikan gambaran singkat tentang penelitian yang dilakukan sebelum menulis penelitian ini. Kajian sebelumnya ini mengutip kajian literatur dan referensi penelitian sebelumnya untuk memberi kesempatan kepada penulis untuk memperluas teori yang digunakan dalam penelitian ini..

Penelitian sebelumnya dilakukan oleh M. Reza Pahlawan dan Dodi Putra Yani (2020) dengan judul Sistem Informasi *Monitoring* Data Produksi Berbasis *Android* di PT SIIX. Tujuan dari penelitian ini adalah mengembangkan aplikasi android untuk memonitor data produksi yang dapat meningkatkan efisiensi dan kinerja perusahaan. Penelitian ini menerapkan metode SDLC (*Software Development Life Cycle*). Hasil yang didapat dari penelitian ini menghasilkan sebuah aplikasi pelacakan produksi lengkap yang dapat digunakan oleh karyawan [4].

Penelitian sebelumnya pernah dilakukan oleh Masmur Tarigan dan Sandri Alfarisi (2020) dengan judul Membangun Aplikasi Perintah Kerja Berbasis Android *Mobile* PT. Pencetakan gravure modern di Indonesia. Tujuan dari penelitian ini adalah untuk mengurangi keterlambatan proses produksi akibat ketidaktepatan jadwal akibat perubahan data dan lambatnya penyampaian informasi. Peneliti ini menggunakan metode *prototype*. Berdasarkan hasil penelitian ini, karyawan lebih mudah untuk menjaga jadwal karena lebih banyak data yang dapat diterima dan ditampilkan lebih efektif dan efisien, serta dapat mencegah terjadinya pengurangan produksi. [5].

Penelitian sebelumnya dilakukan oleh Putra Ibnu Chajar (2018) dengan judul “Aplikasi Proses Bisnis Menjahit Pakaian Berbasis *Mobile Android*”. Tujuan dari penelitian ini adalah untuk membantu penjahit mempermudah proses bisnisnya dan meminimalisir *human error* yang terjadi, seperti kehilangan data

atau *deadline* yang terlewat, sehingga diharapkan proses bisnis dapat berjalan lebih baik. Metode perancangan yang digunakan adalah metode *prototype*. Berdasarkan temuan penelitian ini, terungkap bahwa hasil yang diperoleh selama proses pengujian sesuai dengan desain sistem aplikasi ini. Dari sini dapat disarikan bahwa aplikasi pengolahan bisnis pakaian yang disesuaikan berbasis perangkat seluler android telah berhasil dikembangkan dan dapat beroperasi dengan lancar. [6].

Penelitian sebelumnya dilakukan oleh R. Wisnu Prio Pamungkas, Budi Pratama Wijaya Putra dan Andy Achmad H. dengan judul Optimalisasi Pengawasan Produksi Menggunakan Metode *Sequencing* Menggunakan Aplikasi Berbasis *Android*. Tujuan dari penelitian ini adalah merancang sistem aplikasi perencanaan produksi yang dapat menampilkan informasi status proses perintah kerja sehingga fungsi kontrol produksi dapat diterapkan secara optimal. Peneliti dalam proses pengembangan aplikasi dikenalkan dengan metode *sequencing* karena dapat digunakan untuk menentukan urutan dan prioritas waktu tugas sehingga tugas-tugas tersebut diproses secara sistematis dan proporsional sesuai dengan urutan dan waktunya. Berdasarkan hasil penelitian dapat disimpulkan bahwa dengan bantuan sistem aplikasi perencanaan produksi, personel produksi dan operator produksi dapat dengan mudah melihat informasi urutan prioritas perintah kerja dan perintah kerja melalui sistem pencarian, sehingga tindakan pengendalian dalam produksi menjadi lebih optimal [7].

Penelitian sebelumnya pernah dilakukan oleh Hendy Tri Wahyu Ardiansyah (2022) dengan judul Merancang dan Membangun Sistem Penjadwalan Khusus Berbasis *Android* Menggunakan Metode *Prototype*. Tujuan dari penelitian ini adalah menyesuaikan aplikasi perangkat lunak secara otomatis dengan memprioritaskan beban kerja kecil terlebih dahulu. Penelitian ini menggunakan metode *Prototype*. Hasil dari penelitian ini adalah aplikasi perencanaan berjalan dengan sukses dan berjalan dengan baik [8].

Penelitian sebelumnya dilakukan oleh Abdul Rokhim dan Muhammad Abdul Rizki (2021) dengan judul Penerapan Metode Prototipe Pada Aplikasi Perencanaan Kerja dan Kegiatan Pegawai Berbasis PWA. Tujuan dari penelitian

ini adalah untuk membuat aplikasi penjadwalan pekerjaan beserta fungsinya. Penelitian ini menggunakan metode prototipe. Hasil dari penelitian ini adalah aplikasi telah bekerja dengan baik dan memiliki fitur *push notification* yang mempermudah jadwal kerja karyawan [9].

Penelitian sebelumnya pernah dipublikasikan oleh Rodison Malau, Agustian Suseno, dan Wahyudi (2022) dengan judul Merancang Sistem Informasi Produksi Berbasis Web Menggunakan Metode *Prototyping* di PT. Aisyah Berkah Utama. Riset ini memudahkan pemilik AMDK untuk secara otomatis mendeteksi tingkat produksi harian tanpa melihat secara langsung. Penelitian ini menggunakan metode *prototype*. Hasil penelitian ini menunjukkan bahwa aplikasi berjalan lancar dan dapat membantu pengusaha mengelola data produksi secara efektif [10].

Tabel 2.1 Penelitian Terdahulu

No	Judul	Tahun	Metode	Masalah	Hasil	Perbedaan
1	Sistem Informasi <i>Monitoring Data</i> Produksi Berbasis Android Di PT SIIX	2020	SDLC (<i>Software Development Life Cycle</i>)	Karyawan masih melakukan pengolahan data produksi menggunakan MS Excel sehingga lamanya proses <i>input</i> data, resiko kehilangan data.	Aplikasi ini memberikan informasi kepada karyawan tentang data data produksi di PT SIIX.	Penelitian ini menggunakan metode SDLC sedangkan peneliti menggunakan metode <i>Prototype</i> .
2	Membangun <i>Mobile Apps Job Order</i> Berbasis Android Pada PT.Modern Gravure Indonesia	2020	<i>Prototype</i>	Produksi mengalami penundaan dalam proses produksi karena perubahan data dan penjadwalan yang tidak tepat yang disebabkan oleh pengiriman informasi yang lambat.	Memfasilitasi karyawan untuk mendapatkan jadwal kerja karena informasi dapat diterima dan ditampilkan dengan lebih mudah, efektif, dan efisien, serta dapat mengurangi angka hasil produksi yang tidak	Penelitian ini hanya membuat aplikasi penjadwalan pekerjaan sedangkan peneliti membuat aplikasi tahapan produksi

No	Judul	Tahun	Metode	Masalah	Hasil	Perbedaan
					optimal karena dampak keterlambatan.	
3	Aplikasi Proses Bisnis Penjahitan Pakaian Berbasis <i>Mobile</i> Android	2018	<i>Prototype</i>	Proses penjahit masih dilakukan secara manual mulai dari pemesanan hingga penyelesaian	Aplikasi dapat memudahkan pelanggan memesan penjahit.	Penelitian ini membuat aplikasi khusus proses penjahitan pakaian sedangkan peneliti membuat aplikasi tahapan produksi
4	Optimalisasi Pengawasan Produksi Dengan Metode <i>Sequencing</i> Menggunakan Aplikasi Berbasis	2019	<i>Sequencing</i>	Kegiatan pemantauan terhadap status proses produksi setiap <i>work order</i> yang sedang berlangsung masih dilakukan secara manual, yaitu dengan mengunjungi setiap mesin.	Mempermudah karyawan untuk melihat informasi urutan prioritas pengerjaan pesanan kerja, melihat daftar tunggu pesanan kerja melalui sistem aplikasi	Penelitian ini menggunakan metode <i>Sequencing</i> sedangkan peneliti menggunakan metode <i>Prototype</i> .

No	Judul	Tahun	Metode	Masalah	Hasil	Perbedaan
	Android					
5	Rancang Bangun Aplikasi Sistem Penjadwalan Penjahit (<i>Tailor Scheduling</i>) Berbasis Android Menggunakan Metode Prototyping	2022	<i>Prototype</i>	Pencatatan pesanan yang datang masih ditulis pada nota sehingga rawan hilang.	Mendapatkan penjadwalan otomatis dengan mengutamakan jadwal dengan tugas ringan	Penelitian ini membuat aplikasi penjadwalan otomatis sedangkan peneliti membuat pencatatan tahapan produksi.
6	Penerapan Metode <i>Prototype</i> Pada Aplikasi Penjadwalan Kerja Dan Kegiatan Karyawan	2021	<i>Prototype</i>	Menggunakan Ms. Excel untuk mengelola data	Berhasil membuat aplikasi penjadwalan kerja dan memiliki fitur <i>push notification</i> untuk memberitahu karyawan	Penelitian ini membuat penjadwalan kerja dan kegiatannya sedangkan peneliti membuat

No	Judul	Tahun	Metode	Masalah	Hasil	Perbedaan
	Berbasis Pwa					pencatatan tahapan produksi.
7	Perancangan Sistem Informasi Produksi Berbasis Web Menggunakan Metode <i>Protoyping</i> Pada Pt. Aisyah Berkah Utama	2022	<i>Prototype</i>	Pemantauan hasil produksi harus datang langsung ke lokasi produksi.	Pengelolaan data produksi menjadi lebih optimal dan produktif	Penelitian ini membangun aplikasi monitoring produksi sedangkan peneliti membuat aplikasi pencatatan tahapan produksi.

Dari tujuh penelitian yang telah diuraikan, terdapat perbedaan dengan penelitian yang akan dibuat yaitu metode dipakai pada penelitian diatas berbeda dengan metode yang peneliti pakai yaitu *Prototype*. Pada beberapa penelitian terdapat metode *Prototype* namun studi kasus yang digunakan terdapat perbedaan dengan aplikasi pencatatan tahapan produksi.

2.2 Dasar Teori

2.2.1 Android

Android merupakan *platform* perangkat bergerak. Android tidak membedakan antara aplikasi inti maupun aplikasi eksternal sehingga aplikasi dapat memakai perangkat keras, termasuk akses ke data ponsel atau data sistem itu sendiri. [11].

Android adalah nama sistem operasi di perangkat seperti komputer, tablet, *smartphone*, dan ponsel. Sistem operasi yang terdapat pada *android* berbasis *Linux* yang dikembangkan oleh *Google Inc.* Ikon Android memiliki ciri khas berbentuk robot dengan dua antena di kepalanya, melambangkan bahwa Android adalah OS canggih untuk perangkat dan telepon pintar. Sejak dirilis tahun 2007, Android beberapa kali memperbarui versinya. Versi Android memiliki nama unik yang kebanyakan merupakan nama makanan manis [12].

2.2.2 Metode *Prototype*

Model prototipe adalah teknik yang dapat digunakan untuk mencari informasi khusus dengan cepat tentang apa saja yang dibutuhkan pengguna. Fokus menyampaikan area perangkat lunak yang terlihat oleh pelanggan atau pengguna. Pelanggan/pengguna mengevaluasi prototipe dan digunakan untuk memverifikasi persyaratan perangkat lunak. Prototipe didefinisikan sebagai alat yang memberikan gambaran kepada pengembang dan pengguna tentang bagaimana suatu sistem akan bekerja dalam bentuk lengkapnya. Proses pembuatan desain menggunakan metode ini disebut *prototyping* [13].

Pada model *prototype*, desain dibuat dan dipresentasikan kepada pelanggan kemudian diberi kesempatan untuk berkontribusi agar perangkat lunak yang akan dibuat dapat memenuhi keinginan dan kebutuhan. perbaikan dan presentasi *prototype* dilakukan beberapa kali hingga tercapai kesepakatan bentuk perangkat lunak yang akan dikembangkan [14].

Metode *prototype* memiliki tahapan pembuatan yang dijelaskan sebagai berikut:

1. *Communication*
Mengumpulkan data *prototype* yang akan dibuat. Pada penelitian ini yaitu dengan cara melihat proses bisnis yang dibuat dan melakukan wawancara langsung kepada pengguna.
2. *Quick plan*
Tahapan perencanaan kebutuhan mengenai desain dan fitur yang ada. Penelitian ini berupa diagram UML sebagai perencanaan awal.
3. *Quick Design*
Membuat desain sederhana aplikasi. Dapat menggunakan gambar pada kertas maupun aplikasi *mockup*. Peneliti menggunakan *mockup* sebagai desain cepat.
4. *Prototype*
Proses pembuatan desain *prototype* yang nantinya digunakan sebagai tampilan perangkat lunak.
5. *Deployment Delivery & Feedback*
Pada tahap ini aplikasi dibuat berdasarkan hasil akhir *prototype* sehingga dapat dipakai pengguna [15].

Prototype dari sebuah proyek perangkat lunak harus dikembangkan dengan cepat dan sering mengabaikan praktik pemrograman terbaik [16].

2.2.3 Pemrograman

2.2.4.1 Dart

Bahasa Dart adalah salah satu bahasa pemrograman yang dikembangkan oleh *Google*, bahasa tujuan umum yang dapat dipakai untuk mengembangkan berbagai platform seperti *web*, server seluler, dan IOT. Bahasa pemrograman dart juga merupakan bahasa dasar dari *flutter* [17].

Dart merupakan bahasa pemrograman terstruktur sumber terbuka untuk membangun aplikasi android, iOS, maupun *web* berbasis *browser* yang kompleks. Pengguna dapat mengakses aplikasi berbasis Dart baik secara langsung di *browser* yang telah mendukung kode Dart atau dengan mengompilasi kode Dart pengguna ke dalam *JavaScript*. Dart memiliki sintaksis yang familier dan berbasis kelas, diketik secara opsional, dan utas tunggal. Ini memiliki model konkurensi yang disebut *Isolate* yang memungkinkan eksekusi paralel. Selain menjalankan kode Dart di browser web dan mengonversinya menjadi *JavaScript*, pengguna juga dapat menjalankan langsung kode Dart dari baris perintah. Sintaks Dart memiliki kemiripan dengan bahasa pemrograman *Java*, *C#* dan *JavaScript*. Salah satu tujuan utama Dart adalah membiasakan bahasa. Dart memiliki *script* panah kecil yang terdiri dari satu fungsi yang disebut *main* [18].

2.2.4.2 Flutter

Flutter merupakan Framework yang dikembangkan oleh *Google* berbentuk SDK (*Software Development Kit*) yang digunakan untuk mengembangkan aplikasi seluler. Kerangka kerja ini dapat dipakai untuk membangun atau mengembangkan aplikasi seluler yang berjalan di perangkat iOS dan Android. Flutter mudah digunakan pada perangkat *desktop* karena hanya membutuhkan Android SDK dan dapat dikembangkan dengan *editor* dari Android Studio hingga *Notepad*. Flutter memiliki fitur unik bernama *hot reload* yang

memungkinkan Anda memvalidasi perubahan *script* tanpa mengulang atau menjalankan ulang aplikasi yang sedang anda kembangkan [19].

Flutter menawarkan dua jenis *widget* untuk *developer* aplikasi: *Material Design* dan *Cupertino*. *Material Design* merupakan bahasa desain bawaan *flutter* yang dikembangkan oleh *Google* yang identik dengan tema yang digunakan di Android. *Cupertino* atau dengan kata lain *iOS style* adalah bahasa desain yang digunakan oleh *iOS*. Flutter memiliki lebih banyak *widget* Desain Material daripada widget *Cupertino*, tetapi jangan khawatir, widget dapat digunakan di lingkungan yang berbeda pada sistem operasi perangkat yang berbeda [20].

2.2.4 Perancangan Basis Data

2.2.5.1 Firebase

Firebase merupakan layanan BackEnd as a Service (BaaS) yang dikembangkan *Google*. Firebase memudahkan pengembang untuk membangun aplikasi mereka. Layanan Firebase yang digunakan dalam sistem rekayasa mencakup autentikasi yang memungkinkan pengguna untuk masuk dan terhubung langsung ke *server* Firebase. *Firebase Storage* memungkinkan pengembang mengunggah atau mengunduh *file*, dan *Firebase Cloud* bertindak sebagai kerangka kerja tanpa server yang memungkinkan kode berjalan secara otomatis di latar belakang sebagai respons terhadap pemicu fungsional atau permintaan HTTPS. Firebase memiliki pustaka yang luas dan terintegrasi dengan banyak kerangka kerja lain seperti NodeJS, Java, Javascript, ReactJS, dan lainnya [21].

2.2.5.2 UML

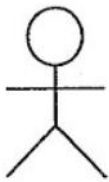
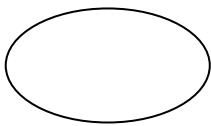

Unified Modeling Language atau biasa disingkat UML merupakan bahasa berbasis grafik atau gambar untuk memproyeksikan, mendefinisikan, membuat, dan mendokumentasikan sebuah sistem perangkat lunak OO (Object Oriented). UML sendiri



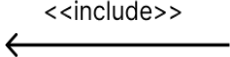
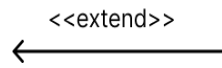
juga menyediakan standar desain sistem yang mencakup konsep proses bisnis, menulis kelas dalam beberapa bahasa pemrograman, skema basis data, dan komponen yang dibutuhkan dalam sistem perangkat lunak [22]. Terdapat berbagai macam diagram UML yang biasa diterapkan dalam proses pengembangan sistem, misalnya:


1. Use Case Diagram

Use case diagram model perilaku (*behaviour*) dari suatu sistem informasi yang menggambarkan interaksi antar aktor sistem tersebut. *Use case diagram* memvisualisasikan sudut pandang pengguna dan fokus untuk menggambarkan fungsi dari sistem [23]. Simbol yang terdapat pada *use case diagram* dapat dilihat pada tabel berikut:

Tabel 2.2 Simbol Use Case Diagram

Gambar	Nama	keterangan
	Aktor	Merepresentasikan orang, proses, atau sistem yang berhubungan dengan sistem operasi.
	<i>Use Case</i>	Entitas bertukar informasi dengan entitas lain maupun aktor.
	<i>System</i>	Sistem mengidentifikasi paket yang mewakili sistem dengan cara terbatas.



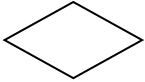


Gambar	Nama	keterangan
	<i>Dependency</i>	<p>Proses menjalin hubungan dengan unsur-unsur independen, dan kemudian mempengaruhi elemen-elemen lain yang tidak bergantung pada elemen-elemen independen tersebut.</p>
	<i>Generalization</i>	<p>Hubungan generalisasi dan spesialisasi (umum-khusus) dimana salah satu elemen merupakan bentuk umum dari elemen lainnya</p>
	<i>Include</i>	<p>Menunjukkan bahwa satu <i>use case</i> sepenuhnya mendemonstrasikan fungsionalitas <i>use case</i> lainnya.</p>
	<i>Extend</i>	<p>Kasus penggunaan target memperluas perilaku sumber sampai batas</p>

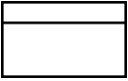
Gambar	Nama	keterangan
		tertentu.
	<i>Association</i>	Menggambarkan hubungan antara aktor dan <i>use case</i> .

2. Activity Diagram

memvisualisasikan bermacam *flow* aktivitas sistem yang direncanakan dari awal aliran aktivitas, proses yang dapat terjadi, dan bagaimana aktivitas tersebut diselesaikan [24].

Tabel 2.3 Simbol Activity Diagram

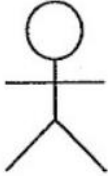
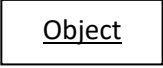

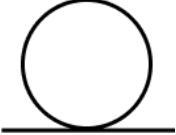
Gambar	Nama	Keterangan
	Titik awal	Awal mula aliran aktivitas
	<i>activity</i>	Menggambarkan aktivitas yang dikerjakan sebuah sistem yang ditandai dengan kata kerja.
	Percabangan	Aktivitas dari aktor maupun sistem yang memiliki dua atau lebih <i>output</i> .
	Penggabungan	Menggabungkan beberapa aktivitas menjadi satu aliran.
	Status akhir	Titik akhir dari aktivitas. Menandakan



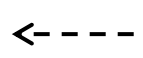
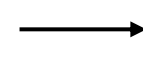
Gambar	Nama	Keterangan
		bahwa aktifitas telah selesai.
	<i>Swimlane</i>	Penanggung jawab terhadap aktifitas yang dimiliki setiap organisasi bisnis.

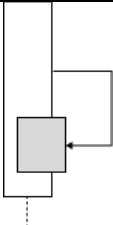

3. Sequence Diagram

Merepresentasikan perilaku objek pada diagram *use case* dengan mendepelitanakan *lifeline* objek maupun pesan yang dikirim dan diterima antar objek [25].

Tabel 2.4 Simbol Sequence Diagram

Gambar	Nama	Keterangan
	Aktor	Menggambarkan orang yang berinteraksi dengan sistem
	<i>Object</i>	Objek yang menggambarkan kelas. Dapat pula dinamai dengan kelasnya.
	<i>Lifeline</i>	Menggambarkan “garis hidup” sebuah <i>object</i> .
	<i>Entity class</i>	Melambangkan <i>database</i> yang berada pada sistem

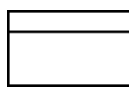


Gambar	Nama	Keterangan
		dan menjadi dasar pembuatan <i>database</i> .
	<i>Boundary class</i>	Merepresentasikan antarmuka yang ditampilkan kepada aktor. Entitas ini berhubungan langsung dengan pengguna.
	<i>Control class</i>	Melambangkan proses yang terjadi di dalam sistem tanpa melibatkan <i>Entity class</i> dan <i>boundary class</i> . contohnya yaitu verifikasi data dan perhitungan
	<i>Return</i>	Simbol mengembalikan data kepada suatu <i>class</i> .
	<i>Message</i>	Simbol mengirim data kepada suatu <i>class</i> .

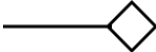


Gambar	Nama	Keterangan
	<i>Recursive</i>	Menggambarkan proses yang memanggil dirinya sendiri.
	<i>Activation</i>	Menggambarkan seberapa lama suatu <i>class</i> berlangsung. semakin panjang aktivasi maka semakin lama pula <i>class</i> dijalankan.

4. Class Diagram

Diagram Kelas (*Class Diagram*) menjelaskan hubungan antar kelas atau sistem, *class diagram* memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem [25].

Tabel 2.5 Simbol Class Diagram

Gambar	Nama	Keterangan
	<i>Class</i>	Simbol atribut dan operasi yang berada dalam satu kelas
	Asosiasi	Hubungan antar kelas yang bersifat umum.
	Asosiasi berarah	Hubungan yang melambangkan bahwa suatu kelas digunakan

Gambar	Nama	Keterangan
		untuk kelas yang lain
	Agresiasi	Relasi antar kelas yang memiliki makna semua bagian.
	Generalisasi	Relasi antar kelas dengan makna generalisasi spesialisasi (umum khusus).
	<i>Dependency</i>	Mengartikan bahwa suatu kelas membutuhkan kelas yang lain.

2.2.5 Pengujian Sistem

2.2.6.1 Black Box

Pengujian *black box* adalah metode pengujian sistem tanpa memperhatikan detail perangkat lunak. Tes ini hanya memeriksa nilai output terhadap nilai input yang sesuai. Tidak ada upaya yang dilakukan untuk mencari tahu kode program apa yang digunakan hasilnya. Proses pengujian *black box* adalah menguji program yang dibuat dengan memasukkan data pada setiap *form*. Tes ini diperlukan untuk menentukan apakah program bekerja sesuai dengan kebutuhan perusahaan [26].

2.2.6.2 User Acceptance Test (UAT)

Pengujian ini bertujuan untuk membuktikan kepada pengguna bahwa sistem bekerja sesuai dengan kebutuhan pengguna.

Jenis tes yang digunakan adalah deskriptif kuantitatif dengan menyebarkan angket untuk menguji objek penelitian melalui aplikasi media pembelajaran interaktif dengan menggunakan metode UAT (*User Acceptance Test*). Pengujian kegunaan didasarkan pada lima komponen, yaitu kemampuan belajar, efisiensi, daya ingat, keamanan pengguna atau pengurangan kesalahan, dan kepuasan..

Learnability mengukur seberapa mudah melakukan tugas-tugas sederhana ketika Anda pertama kali menemukan desain. Efisiensi mengukur kecepatan penyelesaian tugas tertentu setelah mempelajari desain. Recall melihat seberapa cepat pengguna kembali ke desain setelah beberapa saat. Kesalahan melihat berapa banyak kesalahan yang dibuat pengguna, seberapa serius kesalahan itu, dan seberapa mudah mereka menemukan solusi. Kepuasan mengukur kepuasan dengan menggunakan desain [27].